ENHANCING SECURITY IN WIRELESS SENSOR NETWORKS INTEGRATED TO INTERNET OF THINGS USING DIFFERENT SCHEMES

A Thesis Submitted To

Delhi Technological University

for the Award of the Degree of

Doctor of Philosophy

In

Electronics and Communication Engineering

by

GEBREKIROS GEBREYESUS GEBREMRAIAM

(2K19/PHDEC/26)

Under the Supervision of

Dr. JEEBANANDA PANDA

Professor in the Department of Electronics and Communication Engineering

Dr. INDU SREEDEVI

Professor and Dean (Student Welfare)

Professor in the Department of Electronics and Communication Engineering

Department of Electronics and Communication Engineering

Delhi Technological University (Formerly DCE)

Bawana Road, Delhi - 110042, India

September 2023

DELHI TECHNOLOGICAL UNIVERSITY

CERTIFICATE

This is to certify that the doctoral dissertation by Gebrekiros Gebreyesus Gebremariam (Reg. No.: 2K19/PHDEC/26) entitled "**Enhancing Security in Wireless Sensor Networks Integrated to Internet of Things Using Different Schemes**" is based on the author's own original research and meets all requirements for the degree of Doctor of Philosophy at Delhi Technological University. We can attest that his work done under our supervision meets the standards required for the thesis's submission.

It is also guaranteed that the research presented in this thesis has not been submitted, in whole or in part, to any other educational establishment for credit toward another degree.

Prof. J. Panda
Supervisor
Professor
Dept. of ECE.
Delhi Technological University

Prof. S. Indu
Supervisor
Professor
Dept. of ECE
Delhi Technological University

Prof. O.P. Verma
Head of the Department
Dept. of Electronics and Communication
Delhi Technological University

DECLARATION OF AUTHORSHIP

I hereby declare that the work which is being presented in this thesis report **entitled “Enhancing Security in Wireless Sensor Networks Integrated to Internet of Things Using Different Schemes”** in partial fulfilment of the requirement for the thesis of the Doctoral Program in the field of Electronics and communication Engineering submitted to Delhi Technological University, Delhi, India is an authentic record of my work carried out under the guidance of **Prof. J. Panda** and, **Prof. S. Indu** in Department of Electronics and Communication Engineering, Delhi Technological University, Delhi. The matter embodied in this thesis has not been submitted for the award of any other degrees in any institution/university. This report contains no material accepted for the award of any other degree or diploma except where due reference is made. To the best of my knowledge, this work contains no material previously published or written by another person except where due reference is made in the text of the thesis.

Gebrekiros Gebreyesus Gebremariam
Research Scholar
Department of Electronics and communication Engineering
Delhi Technological University,
Delhi-110042, India

ACKNOWLEDGMENTS

I would like to acknowledge and express my gratitude to Prof Jeebananda Panda and Prof Sreedevi Indu for their invaluable guidance and support as my PhD supervisors. Their expertise, dedication, and mentorship have been instrumental in shaping my research journey and enhancing my knowledge and skills in the field. I am truly grateful for their continuous encouragement, insightful feedback, and unwavering belief in my abilities. Their commitment to excellence and passion for research has been a constant source of inspiration throughout my academic pursuit. I extend my sincere appreciation to Prof Jeebananda Panda and Prof Sreedevi Indu for their significant contributions to my development as a researcher and for their continued support in my academic endeavours.

I am grateful to the department of electronics and communication engineering, Delhi Technological University, Delhi, India, for providing the necessary resources and facilities that enabled me to pursue my research. The department of electronics and communication engineering has been a stimulating academic environment, and I am thankful for the opportunities it has provided me to collaborate with fellow researchers and attend conferences and workshops.

I would like to extend my heartfelt appreciation to Professor O.P. Verma, the Head of the Electronics and Communication Department, for his unwavering support and cooperation. Additionally, I am grateful to Professor Neeta Pandey, the DRC Chairman of the Electronics and Communication Department, for her constant encouragement and support throughout my journey.

I would like to express my sincere gratitude to the department heads for electronics and communication during the period of 2019-2023. Their guidance and assistance have been tremendously helpful throughout my journey. Their expertise and dedication have played a significant role in shaping my understanding of the field.

I would also like to extend my heartfelt appreciation to the chairman and the professors of the DRC's Electronics and Communication Department. Their unending encouragement and support have been invaluable. Their passion for teaching and research has been inspiring, and I am grateful for the opportunities they have provided me to expand my knowledge and skills.

I would like to acknowledge the support and assistance of my colleagues and research peers. Their intellectual discussions, sharing of ideas, and camaraderie have been invaluable throughout this journey. Special thanks to colleagues like Enock Osoro Omayio, Rajiv Yadav, Ishu Tomar, Shashank, and all for their collaboration and insightful discussions that significantly contributed to the development of my research.

I am indebted to my friends and family for their unwavering support, understanding, and encouragement during the highs and lows of this undertaking. Their belief in me and their constant motivation have sustained me throughout this challenging process.

Lastly, I want to express my deepest gratitude to the participants of my study who generously shared their time, knowledge, and experiences. Their contributions were essential to the success of my research, and I am truly grateful for their involvement.

In conclusion, I am humbled and grateful to all those who have contributed to the realization of this research and the completion of this thesis. Their support and encouragement have been invaluable, and I am forever indebted to them.

Gebrekiros Gebreyesus Gebremariam
(2K19/PHDEC/26)
Department of Electronics and communication Engineering
Delhi Technological University,
Delhi-110042, India

**This thesis is dedicated to
my parents.**

For their endless love, support, and encouragement

ABSTRACT

This work focuses on enhancing security in wireless sensor networks (WSNs) integrated into the Internet of Things (IoT) by employing different schemes. WSNs are widely used in various fields, such as defence, transportation, healthcare, and environmental monitoring, where they collect and process data from the surrounding environment. However, due to the nature of WSNs and resource constraints, they are vulnerable to security threats. Attacks such as Sybil attacks, routing attacks, and various other forms of malicious activities can compromise the integrity and reliability of WSNs.

The strategies and frameworks proposed in this thesis aim to overcome these security issues. The proposed schemes include robust intrusion detection systems, localization techniques based on range-free and range-based strategies, secure clustering, data aggregation, routing, and optimization techniques. These schemes utilize machine learning models, such as hybrid machine learning algorithms, to detect and classify attacks. The performance of the proposed models is evaluated using benchmark datasets, considering metrics such as training and testing time, precision, recall, F1-score, and accuracy.

The proposed research aims to address security challenges and vulnerabilities present in IoT-based WSNs by employing the design of advanced intrusion detection systems to detect and mitigate both internal and external attacks in IoT-based WSNs. These systems are designed to identify and respond to malicious activities that can compromise the integrity and functionality of the network. Ensemble Machine learning techniques are utilized for classifying and detecting DoS attacks using benchmark datasets.

Integrating trust evaluation methodologies into the localization process helps to solve the security concerns that arise as a result of the procedure. DV-Hop, RSSI, and DE localization techniques assess the nodes' trustworthiness in localization, considering factors such as their behaviour, reliability, and communication history. By evaluating the trustworthiness of nodes, the localization process can mitigate the impact of malicious nodes and ensure the accuracy and integrity of localization results.

The hybrid DV-Hop-RSSI-DE localization approach is coupled with the MLPANN machine learning algorithm and achieves better localization accuracy to detect malicious nodes. MLPANN is trained using labelled datasets to identify patterns indicative of malicious behaviour. The system can detect and classify malicious nodes based on their localization characteristics by leveraging machine learning.

Furthermore, as WSNs are an integral part of the IoT, this thesis also explores security challenges specific to IoT-based WSNs. A hierarchical design incorporating blockchain-based cascaded encryption and trust evaluation is proposed to improve security and service delivery in IoT-WSNs. By combining raw data from devices and identifying risks, federated machine learning improves data security and transport. The proposed approach demonstrates improved performance and security in large-scale IoT-WSNs, leveraging heterogeneous wireless sensor networks to provide secure services.

Secure data aggregation and clustering techniques are also proposed to detect and classify attacks in WSNs. These techniques optimize data aggregation to extend the network lifetime and outperform existing security and performance metrics approaches. These techniques minimize communication overhead and maximize resource utilization while protecting sensitive information using hybrid GA-PSO based on fuzzy rule intelligent routing protocol in WSNs.

Lastly, trust management and routing mechanisms enable secure and efficient communication in distributed and hierarchical network topologies. These mechanisms ensure that data is routed through trustworthy nodes and establish reliable communication paths within the network.

The effectiveness of the proposed schemes and frameworks is validated through simulations and comparisons with recent works. The results confirm the improved security and performance achieved by the proposed methods. By addressing security concerns through intrusion detection, secure localization, machine learning, secure clustering, and trust management, the security and reliability of IoT-based WSNs can be significantly enhanced. The scalability and applicability of the techniques in large-scale deployments are also addressed. This research contributes to developing secure and reliable WSNs integrated into the IoT and provides avenues for future work.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

Abbreviations	Descriptions
BS	Base Station
ANN-IDS	Artificial Neural Network Based Intrusion Detection System
CH	Cluster Head
CNN-MCL	Convolutional Neural Network And Mean Convolutional Layer
D	Distance
E	Energy of the sensor node
DV-H	Distance vector hop
DI-ADS	Deep Intelligent Attack Detection Scheme
ECGAL	energy-efficient clustering and localization based on genetic algorithm
WSN	Wireless sensor networks
RSSI	received signal strength indicator
IoT	Internet Of Things
FEM	Fuzzy Extreme Machines
GBFS-IDS	Gradient Boosting Feature Selection for IDS
HTM-LSTM	Hierarchical Temporal Memory And Long Short-Term Memory
MK-ELM	Multi-Kernel Extreme Learning Machine
LEACH-ANN	Low-Energy Adaptive Clustering Hierarchy Based On ANN
LSTM-FFNN	Long Short-Term Memory and Feed forward Neural Network
ML-ID	Machine Learning Based Intrusion Detection
MLP-ANN	Multilayer Perceptron Artificial Neural Network
PO-CFNN-	Political Optimizer Based On Cascade Forward Neural Network
RL-IDS	Reinforcement Learning Based IDS
RNN	Recurrent neural network
UWB	Ultrawide band
IoT	Internet of things
IDS	Intrusion detection system
MPNN	Multilayer perceptron neural network
IP	Internet protocol
BPNN	Back propagation neural network

FFNN	Feed forward neural network
WSN	Wireless sensor network
DoS	Denial of service
DNN-CSO	deep neural networks with chicken swarm optimization
BS	Base station
Acc.	Accuracy
LEACH	Low energy adaptive clustering hierarchy
CHs	Cluster heads
FPR	False-positive rate
RRSE	Root relative squared error
ROC	Receiver operating characteristics curve
RAE	Relative absolute error
MAE	Mean absolute error
PRC	Precision recall curve
RMSE	Root mean squared error
TPR	True positive rate
TTBM	Time taken to build the model
MCC	Mathew’s correlation coefficient
E_{TX}	Transmitted energy
E_{RX}	Received energy
R_n	Relay node
TDMA	Time Division Multiple Access
WSNs	Wireless sensor networks
ID	Intrusion detection

LIST OF FIGURES

LIST OF TABLE

Chapter 1

1 INTRODUCTION

Wireless Sensor Network(WSN) is the assemblage of homogeneous and heterogeneous resource-constrained sensing devices which sense the environment's physical phenomenon and transmit the information to the sink node (base station) via different modes of communication [1]. Contrarily, the Internet of Things (IoT) is composed of other networked objects which are interconnected to gather, process, refine, and exchange meaningful data over the Internet. These objects are assigned to their respective IP addresses or device identities, and these are able to send and receive data over a network without any human assistance. The information is transferred to the base station for processing as per the requirement of the applications. They are composed of many inexpensive micro sensor nodes deployed in the monitoring area, which is a multi-hop self-organizing network system formed by wireless communication. Nodes are generally low-power and distributed in an ad hoc, decentralized fashion. Although WSNs have gained a lot of popularity, there are some serious limitations when implementing security imposed by resource limitations in memory, computing, battery life, and bandwidth. A range of attacks can target privacy, control, or availability. It has attracted much attention in the international community, involving microwave sensors and micro-mechanics, communication, automatic control, artificial intelligence, and many other disciplines. It integrates sensor technology, embedded computing technology, modern network, wireless communication technology, and distributed information processing technology [2]. Wireless sensor networks (WSN) generally perform network monitoring, collection of information such as network state, data transmission, accurate positioning, and tracking [3].

WSNs are used in many applications, including military, encounter detection, environmental conditions, weather monitoring, object detection, and disaster sensing. Wireless sensor networks consist of transferable and lightweight sensor devices. The sensor devices have the capability of data processing, sensing, and communicating from source node to destination in wireless sensor networks. Sensor nodes also have limited transmission capability and provide the sensed data to the legitimate user. The intermediate node performs the long data transmission process. Due to this, external and internal DoS attacks affect the services and quality of wireless sensor networks. WSNs are suffered from localization, energy consumption, hole coverage, and tolerance to a fault. [8]. The major problem is security is totally not covered and still open. Depending on the deployment scenario and the characteristics of WSNs, the sensors may be vulnerable to a wide range of security attacks [4].

Sensor nodes, base stations, and beacon nodes are the primary building blocks of WSNs. The sensing data is gathered from sensors placed in the surroundings in an ad hoc fashion [1], [6], [7]. Figure 1.1 depicts the two main architectures that the sensor nodes might take on: the distributed architecture and the hierarchical design. The distributed system is made up of sensor nodes and central hubs. Data is gathered by the sensor node and transmitted to the main station. Raw data sensed and collected sent to the cluster head was stored and analyzed at the base station [8]. There are sink nodes, cluster leaders, and cluster members in a hierarchical design. Information is sent to the CH, which then relays it to the BS [9]. The sink nodes are the gateway to the network, where the information from the various sensors is gathered. It is possible to apply the hierarchical design to wide-area network topology.

Wireless sensor networks (WSNs) [10] comprise a collection of hierarchically micro-sensor nodes spread out in the field and connected via multi-hop wireless communication technologies. The sensor nodes contain sensors, radio transceivers, memory chips, and CPUs. When it comes to networks, wireless sensor networks (WSNs) are constantly keeping tabs on things like data transmission and the current state of the network. They keep an eye on things like location and movement [3]. Sybil attacks, wormhole attacks, eavesdropping, and other similar threats [11] are all possible with WSNs.

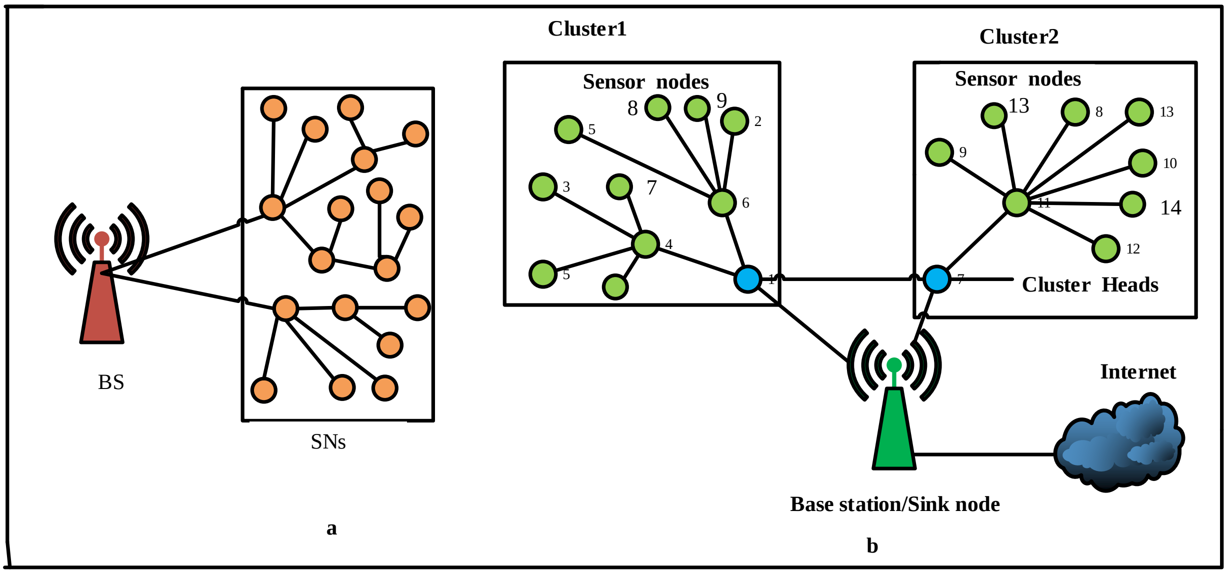


Figure 1.1. The architecture of hierarchical wireless sensor networks (a) and Distributed Wireless Sensor Networks (b) adopted from [1].

Sybil attacks, in which numerous false identities are created and devalued before being used in a coordinated assault on the real node to degrade service, are the most damaging routing attack [4-5]. This issue is fundamental to the study of wireless sensor networks. Still, researchers in other disciplines have also found it useful for examining related topics like architecture, healthcare, disaster management, deployment quality of service, calibration [14], and synchronization.

Wireless sensor networks can monitor the environment using seismic, magnetic, thermal, visual, infrared, radar, and acoustic sensors with low sample rates. The primary functions of sensor nodes are continuous sensing, event identification and detection, and local control of actuators. Figure 1. 2 demonstrates that wireless sensor networks are useful in various settings, including medical, military, environmental, residential, and commercial [15].

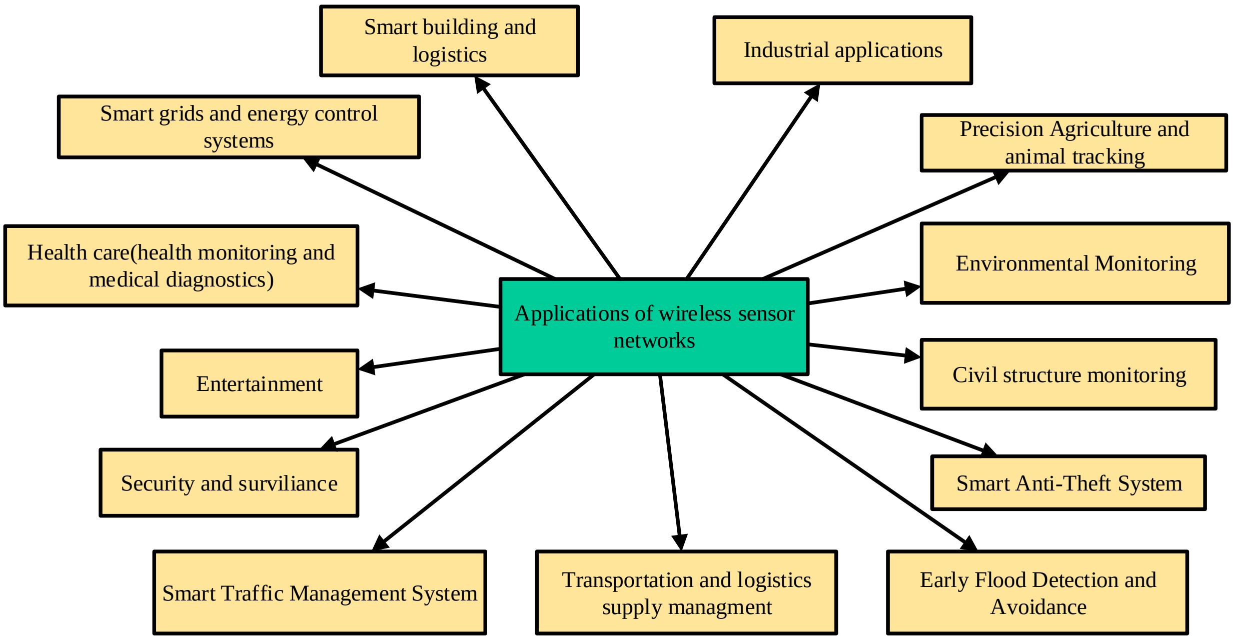

Smart building and logistics
Industrial applications
Smart grids and energy control systems
Precision Agriculture and animal tracking
Health care(health monitoring and medical diagnostics)
Environmental Monitoring
Applications of wireless sensor networks
Entertainment
Civil structure monitoring
Security and surviliance
Smart Anti-Theft System
Smart Traffic Management System
Transportation and logistics supply managment
Early Flood Detection and Avoidance

Figure 1. 2. Wireless sensor network applications in different domains and disciplines.

These networks are used in environmental trackings, such as forest detection, animal tracking, flood detection, forecasting, and weather prediction, and commercial applications like seismic activity prediction and monitoring. Military applications, such as tracking and environment monitoring surveillance applications, use these networks. The sensor nodes from sensor networks are dropped to the field of interest and are remotely controlled by a user. Enemy tracking and security detections are also performed by using these networks. Health applications, such as Tracking and monitoring patients and doctors, use these networks. The most frequently used wireless sensor network applications in Transport systems, such as monitoring traffic, dynamic routing management, parking lots, etc., use these networks.

In an era of unprecedented technological advancement, the Internet of Things (IoT) has emerged as a transformative force, seamlessly interconnecting devices and enabling various applications across various domains. At the heart of this interconnected ecosystem lies the wireless sensor network (WSN), a fundamental component responsible for collecting and transmitting critical data in real time. However, as the IoT landscape continues to expand, so does the vulnerability of these networks to a multitude of security threats. The primary goal of the proposed research thesis is to address these pressing security concerns by leveraging advanced techniques in secure localization, trust evaluation, and intrusion detection.

This thesis focuses on a holistic approach to enhance the security of WSNs within the IoT paradigm. The research endeavors to tackle the problem from multiple angles, starting with developing secure localization techniques. Accurate localization is crucial for not only ensuring the integrity of data but also for enabling efficient and context-aware decision-making within IoT applications. Trust evaluation mechanisms will complement secure localization, aiding in identifying trustworthy nodes and mitigating the risks associated with compromised or malicious sensors. Furthermore, this thesis will delve into designing and implementing robust and scalable intrusion detection systems (IDS) within WSNs. IDS plays a pivotal role in identifying and responding to various types of attacks that can threaten the network's security. Employing machine learning methods against benchmark datasets, the research aims to create IDS systems that can adapt to evolving threats and maintain high levels of accuracy.

The proposed research will also explore secure clustering and intelligent routing methods to provide a comprehensive defense. These techniques will not only aid in detecting attacks but also in localizing them at different network layers, thus preventing their propagation and minimizing their impact.

In conclusion, this thesis endeavors to contribute significantly to IoT-embedded WSN security. By addressing the critical challenges of secure localization, trust evaluation, and intrusion detection, machine learning techniques it aims to create a more resilient and secure foundation for the IoT ecosystem. In doing so, it will help unlock the full potential of IoT applications while safeguarding the data and operations that underpin them.

1.1 IoT-based Wireless Sensor Networks

The development of new technologies and improved network infrastructure have helped the Internet of Things (IoT) expand, resulting in several beneficial effects on society and the economy [16]. Systems, devices, applications, and people who work together to process data are all part of the Internet of Things (IoT) [17]. The Internet of Things enables inanimate things to interact with one another, exchanging data and performing functions such as decision-making and coordination. As a result, these inanimate objects may now see their surroundings, form complex thoughts, and take meaningful actions [18]. The military, smart cities, healthcare, and environmental applications have all made IoT-WSNs a hot topic of study in recent years [19]. Wireless sensor networks (WSNs) have contributed significantly to the development of the IoT and are, therefore, of great academic and practical significance. Wireless sensor networks (WSNs) are ad hoc networks composed of a swarm of devices, or "sensor nodes," each equipped with sensing technologies. Self-organizing, randomly deployable, fault-tolerant sensor nodes must closely monitor a large area. The Internet of Things-WSNs is the way to go when gathering information about the world. The sensor nodes collaborate and review one another to ensure reliable data transmission, identify and eliminate malicious nodes in the network, and improve the quality of the data collected. Malicious assaults can come from within or without, and if they succeed in compromising the beacon node in the cluster area, they can use that node to exploit data and send incorrect routing instructions to the sensor nodes and the base station. The rogue node reduces network efficiency by spreading false information and sowing distrust between the other nodes and the end users.

In order to identify and analyses potentially harmful nodes in a network, the trust evaluation procedure constantly monitors threshold and signal strength values between sensor nodes; due to the energy constraints of IoT devices, mistrust between them is prevalent. So, there is a chance that private data on these devices could be leaked or stolen. Blockchain's unparalleled immutability makes it possible for users to make transactions without verifying the sincerity of their counterparts. As a backup measure, the blockchain keeps a hash of the data from the previous block in each successive block. Having entirely undetectable devices is often a must as well. The majority of devices, however, have a simple architecture that cannot handle the complexity of common encryption techniques. Transactions are the primary data transfer unit;

IoT-WSNs entities can establish connections. Every information recorded about a customer's business dealings is regarded as a transaction in the initial blockchain implementation.

Bitcoin networks rely on blockchain, a distributed ledger technology, to maintain their security [20]. This revolutionary structure is widely used in many distributed situations, such as healthcare and automotive Adhoc networks. Many studies have found that blockchain technology is particularly well-suited to tackling security challenges related to the Internet of Things (IoT). Every node in a blockchain network shares responsibility for storing and processing data related to the technology. Distributed blockchain technology has several advantages over centralized management systems, such as reducing the cost of maintaining many Internet of Things devices and eliminating the possibility of a single point of failure.

1.1 Security Requirements and Threats

Protecting the privacy and integrity of data in WSNs is a crucial but difficult task [21]. To ensure the safety of wireless sensor networks, specialized tools and methods are required. Since the nodes' resources are finite, it is critical to employ security mechanisms to maximize the network's safety and longevity [22]. Sensor networks must meet some conditions to guarantee confidential exchanges of information. The four cornerstones of network security for WSNs—availability, secrecy, integrity, and authentication—are essential. The secondarily required features are source localization, self-organization, and data freshness. This is necessary to prevent assaults on data sent over a sensor network [23].

Data Confidentiality: The security of sensitive information is especially at risk in sensor networks due to the high number of intermediary nodes involved in data transmission. The encrypted information is secure because only the recipient can decrypt it back into plaintext.

Data Integrity: It is important that the information received by the receiver has not been tampered with in any way or that the data's integrity has been compromised. An unauthorized party or a hostile environment modifies the original data. The hacker alters the information to suit its purposes and then relays the modified data to the target.

Data Authentication: Data authentication is the process of verifying the identity of a communication node. The receiving node must check each data received to ensure it came from a trusted source.

Data Availability: When data is available, services continue functioning even when under assault, such as from a DoS.

Source Localization: Some programs relay data using the sink node's position data, allowing for precise source localization. Security of location data is crucial. A hostile node can manipulate unencrypted data by providing distorted signal intensities or replaying signals.

Self-Organization: since there is no centralized infrastructure in a WSN, each node functions autonomously. This allows it to respond to environmental changes while retaining its ability to self-organize and self-heal. It's a major obstacle to protecting sensitive information in WSN.

Non-repudiation: This method ensures that a receiving party will not contest the integrity of a communication. It provides evidence of the data's authenticity and reliability. Because of this, disputing a message's origin and its contents' validity is extremely challenging.

Authorization: To ensure that only legitimate nodes (here, sensors) in a WSN or IoT-based communication environment can share data, the property of "authorization" is essential.

Data Freshness: Data freshness refers to the fact that every bit of information sent across the network is completely original. It ensures that the old messages can't be re-sent by any node. Due to this, a time-based counter can be added to verify that the data is up-to-date.

Availability: In WSN and IoT-based communication systems, this feature ensures that authorized users can still connect to the network and use the related services, even if a "Denial-of-Service" (DoS) attack is launched against them.

Forward secrecy: When a device or party exits the WSN and IoT-based communication environment, it should no longer have access to future communications.

Backward secrecy: Messages sent and received in a WSN and IoT-based communication environment must be kept secret from any new devices or parties that join the network.

1.2 Classification of Security Threats

Wireless sensor networks are vulnerable to various internal and external attacks that affect the structure and configuration of the five layers: the physical layer, data link layer, network layer, transport layer, and application layer [1]. Attacks can also be roughly categorized based on capability, routing, and protocol layer. They can attack the radio transmission by adding their data bits to the channel, replaying old packets, and any other type of attack. A secure network ought to support all security properties [23]. Attackers can compromise a network by either deploying malicious nodes with identical capabilities or by capturing legitimate nodes and replacing their memory with malicious data.

Figure 1. 3 displays attacks corresponding to the different WSNs stack layers. The network layer attacks are the most dangerous of the several types shown in Figure 1. 3 because they destabilize the entire network and make it more vulnerable to DDoS attacks by disrupting its routing system. The layer-by-layer breakdown simplifies identifying threats, devising defences, and examining layer-specific properties [24], [25]. Analysis of the critical issues of these attacks and defences provides a condition for developing attack detection and classification model enhancing security in WSNs. A passive or active adversary may employ the following attacks against the WSN and IoT-based communication environment [1]: Sniffing or snooping are other names for eavesdropping. It occurs when an outsider listens to a conversation between two or more people. It's also a possible risk to communication using WSNs and the Internet of Things. An attacker intercepts messages and then analyses them to determine who is talking to whom. For a Replay and Impersonation Attack to occur, an adversary must first intercept the messages being sent and received and then deliberately delay or retransmit them. To impersonate another person is to commit an act of malice.

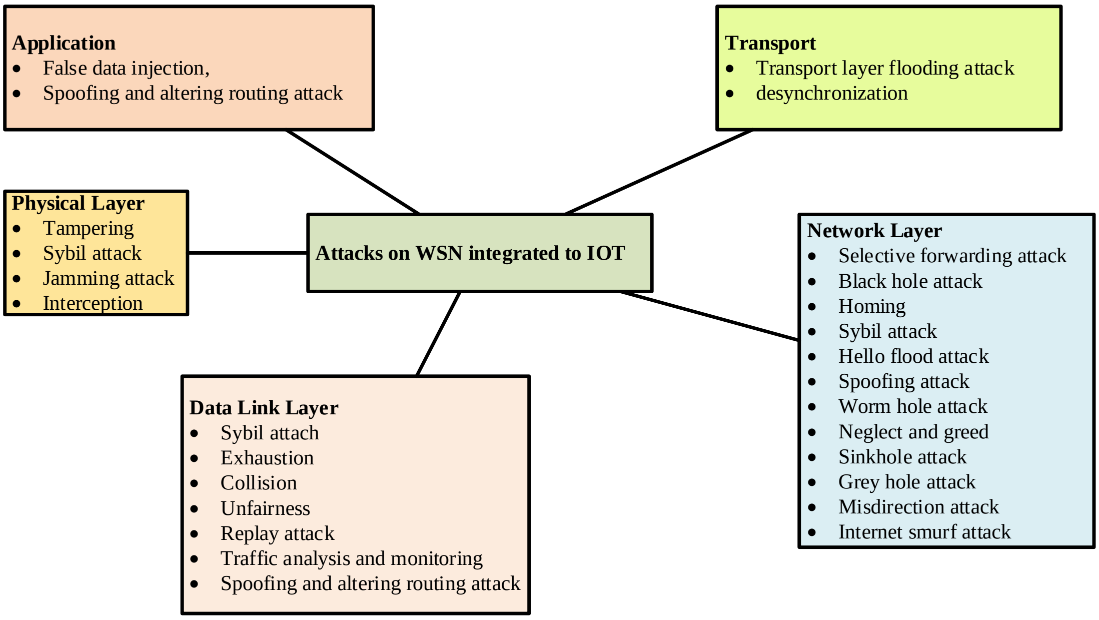


Figure 1. 3. Classification DoS attacks in wireless sensor networks in different layers.

Suppose an adversary can determine the identity of a legitimate communicating party in the network. In that case, they can modify the messages the party has communicated and transmit them on the recipient's behalf. In this malicious conduct, an adversary sits in the middle of a communication channel and attempts to edit, add, or remove information from messages before sending it to their intended recipients.

Attacks in wireless sensor networks can be classified as active or passive based on functional operations and layers of communication protocols [26].

1.2.1 Blackhole Attack

The intruder performs suspicious activities using loopholes for route discovery [27]. It is a common and famous attack in WSN, compromising the router for different reasons. Packets are dropped regularly from unstable networks due to attacks. The suspicious router makes this attack drop packets to target nodes in the network. Blockhole attack customizes the setting of the nodes to drop packets and cannot forward to the target. The information from the blackhole zone is dropped. Figure 1. 4 shows the packet dropping due to a malicious node's presence. The malicious node M sends a fake routing request to Node N1 rather than N4. So node M receives all information from node N1 and sends the packets to another malicious node. Blackhole attacks in WSNs include capturing and reprogramming a set of nodes so they refuse to pass packets along to the sink node [28], as shown in Figure 1. 4 clusters 1. The attacker compromises the information that enters the blackhole region, undermining the performance and lifetime of the network by making partitions.

1.2.2 Sybil Attacks

The Sybil can get the identities by forging its own identities and spoofing the identity of the legitimate in WSNs, as shown in Figure 1. 4 clusters 2. It affects and degrades the overall network security and performance by generating multiple fake identities. Advanced detection techniques are applied for the detection of Sybil attacks. The fuzzification technique detects a Sybil attack using a Multilayer Perception Neural Network (MPNN) to separate the legitimate node and Sybil attack. The nodes develop tables containing local ranging calculations and estimations between neighbour nodes. Assume that the distance from the node [29] $n_a$to the node $n_{b,}$ is d_{ab}^{n}, So that distance detection with e error is given by the following equation (1.1) for all neighbor list nodes b,c $\neq$ a, 1≤b is

$$d_{ab}^{e} = \begin{cases} |d_{ab}^{n} \text{-} d_{ac}^{n}| < e, & \text{false alarm else} \\ |d_{ab}^{n} \text{-} d_{ac}^{n}| \geq e, & \text{normal traffic data} \end{cases} \tag{1.1}$$

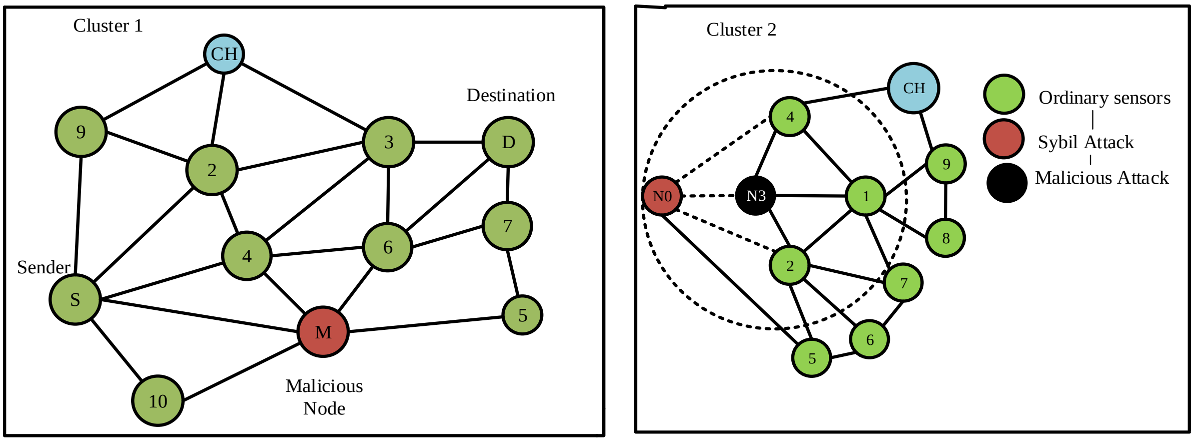


Figure 1. 4. Illustration of a black hole and Sybil attack used to examine lost packets and discovered paths in WSNs.

1.2.3 Wormhole Attack

Wormhole attacks are specific routing attacks in wireless sensor networks by creating a tunnel between two malicious nodes for its functionality. A malicious node absorbs data packets from one coordinated point and passageway it to the other assault node at a specific point in the network. The tunnel can be demonstrated in various paths, including out-of-band channels, high-power transmission, and data packet encapsulation [29]. Figure 1.5 (a) shows a wormhole tunnel that can be used between two malicious nodes through them. The wormhole attack influences the data aggregation, clustering protocols, and location of nodes in WSNs. Wormhole Resistant Hybrid Technique (WRHT) can detect the wormhole attack using a fuzzification approach with Feed Forward Neural Network (FFNN). The WRHT scheme is based on Watchdog and Dolphin. The watchdog and Dolphi are based on the network's packet drop and hop count. WRHT uses PLP and TDP with dual-mode detection techniques to find received packet loss so that wormhole attack works in the encapsulation mode. The formulas for the presence of a wormhole attack are depicted in equations (1.2(1.3):

$$TDP_{P}=1-\left(\prod_{j=1}^{n}(1-TDP_{j})\right) \tag{1.2}$$

$$PLP_{P}=1-\left(\prod_{j=1}^{n}(1-PLP_{j})\right) \tag{1.3}$$

Where TDP_i and PLP_j are time delay probability and packet loss evaluated at node j. The path of the tunnel is defined by the probability of wormhole presence (PWP) as follows using equation (1.4):

$$\mathrm{PWP}_p = \mathrm{TDP}_p + \mathrm{PLP}_p - (\mathrm{TDP}_p \text{ and } \mathrm{PLP}_p) \tag{1.4}$$

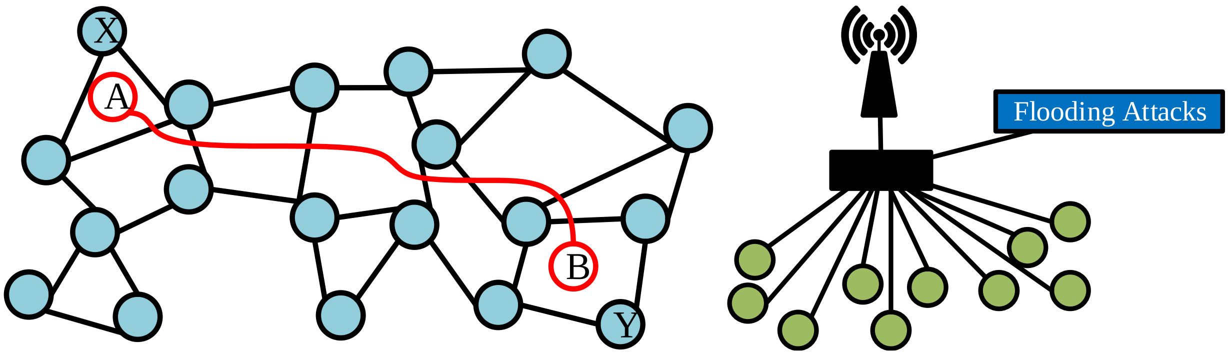
The calculated values move to the FFNN. The fuzzy interface uses anomaly and misuse detection techniques for estimating the wormhole attack using the fuzzy detector module. The detector module detects and stores the malicious node as suspicious.

1.2.4 Flooding Attack

The hello flooding attacks work on assault nodes that disseminate hello packets introducing high-power sensors. The sensor nodes obtain hello packets that they consider incorrectly in the transmitter's received signal strength (RSS). These packets are dropped before reaching the end nodes, as in Figure 1.5 (b). The distance of the nodes is estimated using the equation (1.5):

$$D=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2} \tag{1.5}$$

Where (x_1, y_1) and (x_2, y_2) are the coordinate points of the destination node and cluster head for receiving and advertising the Hello flood traffic. The received signal strength and the distance are moved to the backpropagation neural network (BPNN) using the threshold values. The fuzzy interface detector module uses misuse and anomaly detectors for estimating the Hello flood attack using the AH-IDS. AHIDS utilizes the fuzzy based on MPNN containing BPNN and FFNN supervised machine learning techniques to detect and classify various attacks.



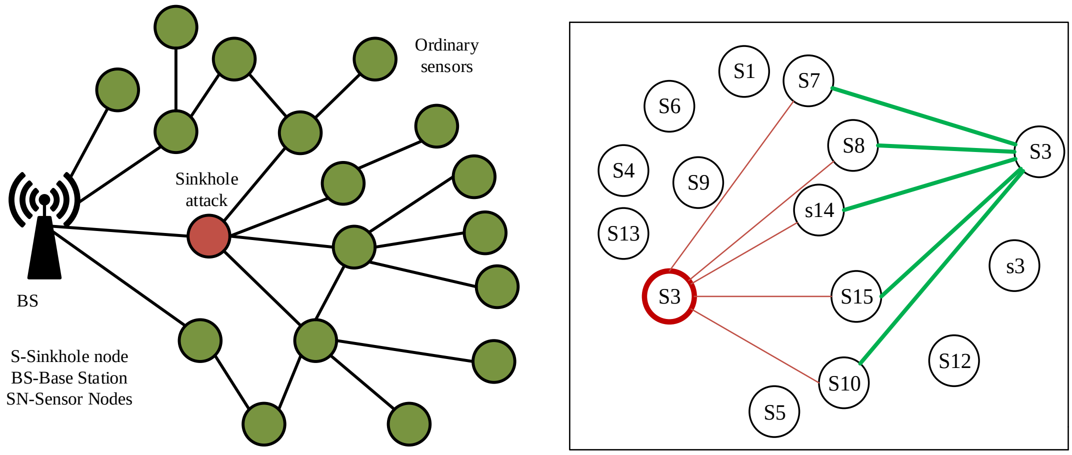
(a) wormhole attack

(b) flooding attack

Figure 1.5. Illustration of wormhole tunnel and Hello flooding attacks in wireless sensor networks.

1.2.5 Sinkhole Attacks

Sinkhole attacks are malicious nodes that advertise the best possible route to the base station and misguide the user frequently [28]. This attack causes a serious threat to wireless sensor networks. The sinkhole attacks use a compromised node for launching attacks that advertise and misguide its neighbours. So that the neighbours forward the traffic using the advertised route. The routing path to the sinkhole attack affects and captivates other nodes closer to it. Figure 1. 6 (a) illustrates the sinkhole attack's graphical illustration.



(a) illustration of sinkhole attack in WSN (b) Clone attack analysis in WSN [30].

Figure 1. 6. Illustration of the sinkhole and cloning attacks in WSNs.

Malicious nodes capture data traffic packets and utilize secret tunnels to send packets to the other colluded node. The tunnel in wormhole attack is used to conduct the sinkhole attack in the network. Colluded nodes deliver data packets to the sink node. The distance between two ends of the wormhole tunnel is compared with other routing paths. They can also prevent multiple routing discoveries of hops from the base station, as in Figure 1. 6(a).

1.2.6 Clone Attacks

Clone attacks are the most vulnerable in WSNs, as shown below in Figure 1. 6 (b). Hybrid techniques are used to detect clone attacks in WSNs [30]. Distributed and centralized techniques are applied for clone attack detection using compressed sensing-based identification, hierarchical node replication attacks detection, real-time clone attack detection, clone node detection, random key pre-distribution, fast detection of replica node, sequential analysis, and efficient and distribution protocol in wireless sensor networks. The keys are predistributed to the various nodes before the wireless sensor nodes are deployed. The nodes have a secret key for building the network after deployment. The generated secret keys are placed in sensor nodes to establish connecting paths and create a graph. The keys are generated randomly for determining mutual connectivity and pool key scheme analysis.

1.2.7 Jamming Attacks

Jamming attacks that target signal transmission are a major concern for WSNs due to the open nature of wireless networks [31]. Jamming attacks are a subset of DoS attacks in which rogue nodes disrupt legitimate nodes' ability to communicate with one another by emitting interference signals. Because they rely on radio waves to communicate data, wireless sensors are vulnerable to signal jams from various sources, compromising the security of transmitted data and disrupting network transmissions. There are many ways in which sensor networks inside the jammers' radiation range can be negatively affected by these attacks. Hence, it has become an essential challenge to figure out how to swiftly detect jammers when they exist and properly determine their attack areas.

In wireless sensor networks, a jamming attack creates a spectrum of disruption that is roughly circular and focused on the jamming transmitter, as shown in Figure 1.7. A sensor node within the jamming range will not be able to communicate normally with a node outside of the jamming range. Communication nodes located on the periphery of the impacted area will have weakened signals during a jamming attack. However, they can still send some data to nodes located beyond the jammed area.

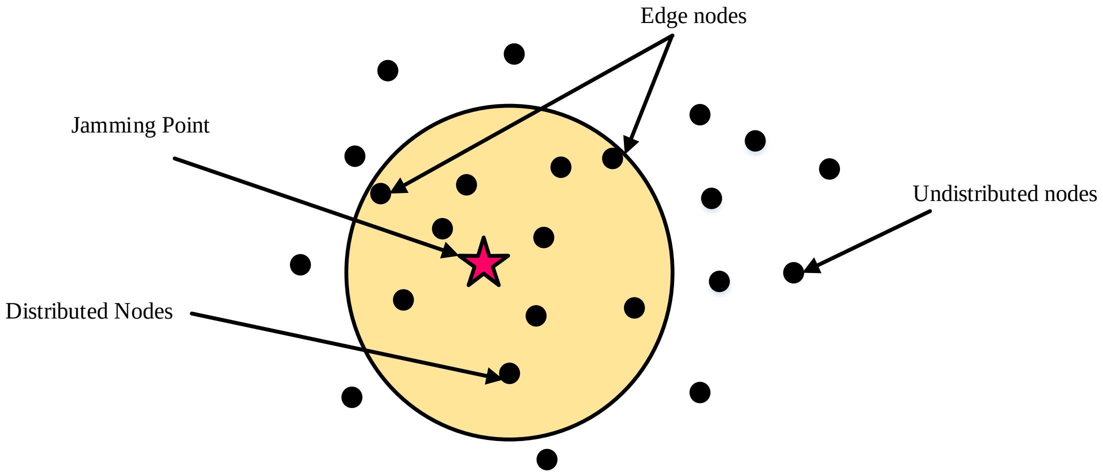


Figure 1.7. Jamming attacks signal in distributed wireless sensor networks [31].

Finding the source of the jamming quickly through these nodes and implementing the appropriate security measure are crucial steps in restoring normal communication in wireless sensor networks

1.3 Research Motivation

The Internet of Things (IoT) trend has introduced several new challenges related to security in wireless sensor networks. Security and privacy are the two important components of any system, and IoT-based wireless sensor networks are no exception. As IoT-based wireless sensor networks are designed to operate in dynamic and hostile environments, the security measures must be strong enough to protect the devices from malicious attacks. Researchers have proposed different security measures to secure IoT-based wireless sensor networks from attackers. These include intrusion detection systems (IDS), secure localization techniques, blockchain techniques, machine learning, hybrid models, secure clustering and data aggregation, and attack detection analysis.

Security in wireless sensor networks is of utmost importance as they are prone to various malicious attacks. The main research objective in this area is to develop efficient and secure communication protocols that can detect, prevent, and respond to malicious attacks. Research in security in wireless sensor networks can be conducted through intrusion detection systems, secure localization techniques, blockchain techniques, machine learning, hybrid models, secure clustering, data aggregation, and attack detection analysis.

As the amount of DDoS attacks in the blockchain IoT ecosystem increases, all blockchain-backed IoT networks will be at increased risk. In order to keep an Internet of Things network safe, it requires a distributed security architecture that makes use of blockchain technology. Using the appropriate analytical tools in a decentralized architecture ensures that a security mechanism can handle the enormous amounts of data produced by IoT devices. Developing an IDS that can differentiate between secure and risky Internet transactions is challenging. It is not well understood how to avert DDoS assaults against mining pools until the matching model has been built for blockchain-enabled Internet of Things wireless sensor networks. In response to these challenges, we provided a hybrid federated machine learning-based approach to identifying and pinpointing the origin of routing attacks in WSNs-IoT through the examination of sample datasets.

The hybrid design model and the mitigation of the node impersonation attack are essential for properly deploying wireless sensor networks. Security and privacy are the two major issues in wireless mesh networks. An impersonation attack is the most severe attack in wireless mesh networks. Hybrid secure communication is designed for the detection of the targeted node. The different literature surveys of the existing systems suggest that hybrid optimized techniques are effective for security and network lifetime in wireless sensor networks. However, there are still drawbacks and challenges in designing and planning to optimize the safety and topology in heterogeneous wireless sensor networks. Secure clustering and data aggregation technique is developed to enhance the network security performance using machine learning classifiers with the appropriate network planning and designing in WSNs. Machine learning classification models are applied in the proposed security system for performance maximization of attack detection accuracy and classification. In this work, we developed a secure data aggregation and clustering in hierarchical WSN using machine learning models.

1.4 Problem Statement

Various security measures must be taken to ensure secure communication in a wireless sensor network integrated with the Internet of Things (IoT). These measures may include data encryption and authentication, secure routing protocols, intrusion detection systems, secure localization, machine learning techniques using benchmark datasets, and secure clustering and data aggregation. It is important to consider the scalability of these security measures due to the limited resources and small network size. Varying network topologies, connection patterns, and potentially malicious activity must be considered to ensure a secure system. Security updates should be regularly applied to the network with a degree of trust in the devices and users.

Furthermore, the performance and network lifetime of the security scheme must be evaluated for scalability and service provisioning. It is also important to consider the physical and environmental constraints when designing secure wireless sensor networks integrated into the IoT. This includes deploying sensors in locations where they cannot be tampered with or accessed by malicious actors. Additionally, it is important to consider the network topology when designing the security architecture. This includes deploying nodes to minimize the risk of malicious activities or data interception. Finally, it is important to consider the scalability of security schemes when designing secure wireless sensor networks integrated into the IoT.

Wireless sensor networks (WSNs) are an important Internet of Things (IoT) component. WSNs monitor physical or environmental parameters such as temperature, humidity, and pressure and detect and track objects. The data collected by the sensor nodes is transmitted over the wireless medium to a base station for further processing. However, due to communication's open and wireless nature, WSNs are vulnerable to various attacks. These attacks can disrupt the network’s operation and cause data loss or corruption. Thus, it is important to detect and localize multiple attacks in WSNs. Several algorithms have been proposed to detect and localize multiple attacks in WSNs. However, these algorithms require a high computational complexity and are not suitable for real-time applications.

Moreover, these algorithms often fail to detect sophisticated attacks. To overcome these limitations, we proposed this research using artificial neural networks (ANNs) for detecting and localizing multiple attacks in WSNs. ANNs are well-known for their ability to learn and detect patterns in data. Thus, ANNs can be used to detect malicious activity in WSNs. This research will investigate using ANNs to detect and localize multiple attacks in WSNs. The performance of the proposed ANN-based detection and localization system will be evaluated using simulated WSNs. The proposed system will be compared with existing algorithms to demonstrate its efficacy and efficiency. This research will provide insight into the use of ANNs for detecting and localizing multiple attacks in WSNs.

Existing literature mostly dealt with isolated attacks that were difficult to localise and detect in WSNs. Therefore, correct node positioning and attack identification necessitate using smart and efficient localization and intrusion detection technologies, which are sought after in deploying wireless sensor nodes. This problem requires the creation and application of an innovative and effective strategy. Wireless sensor networks are susceptible to repeated denial-of-service (DoS) assaults that exploit the network's resources; this thesis presented a security localization and detection strategy using optimized multilayer perceptron artificial neural networks for these attacks. The suggested method employs the ANN method to describe the input and output relationships, making it suitable for a wide range of attack detection and classification tasks. The sink node, cluster head, malevolent nodes, and sensor nodes that comprise a distributed hierarchical clustered topology are also part of the conversation. We also explore the efficacy of the proposed system on benchmark datasets, such as CICIDS2018, UNSW-NB 15, WSN-

DS, and NSL-KDD, with various assessment metrics based on training and testing samples. The data set undergoes batch processing.

1.5 Research Objectives

The primary goal of the proposed research is to improve the security of wireless sensor networks that are embedded in the Internet of Things by using secure localization and trust evaluation techniques, as well as by designing robust, scalable intrusion detection systems (IDS), employing machine learning methods against benchmark datasets, and secure clustering and intelligent routing methods to detect and localize attacks at different layers of the network. The specific objectives of the thesis are bulleted as follows:

1. To design robust and secure intrusion detection schemes to enhance security in hierarchically distributed WSNs.

- It is vital to establish intrusion detection strategies that are both reliable and secure to increase the safety of hierarchically distributed wireless sensor networks (WSNs). This can be accomplished by creating reliable and secure intrusion detection systems.

2. To improve sensor node security and localization accuracy for large and scalable IoT-WSNs based on localization techniques.

- Sensor nodes for large and scalable IoT-WSNs dependent on localization techniques must improve their security and accuracy.

3. To design and analyze a secure hybrid model for targeted malicious nodes in WSN based on a blockchain using federated machine learning techniques.

- This objective focuses on accomplishing the following:

❑ Analyze existing WSN security models and identify weaknesses in current security solutions.

❑ Design a novel secure hybrid model for targeted malicious nodes in WSN based on blockchain technology and federated machine learning.

❑ Develop a prototype of the proposed model to be tested and evaluated.

❑ Evaluate the performance of the proposed model and compare it to existing solutions.

❑ Develop a comprehensive report to summarize the findings and recommendations.

4. To enhance WSNs' security-based secure data aggregation and clustering protocols for hierarchically distributed topology. The performance of the network model against routing attacks trust routing techniques are also analyzed.

1.6 Contribution and Thesis Layout

Various schemes and techniques can be employed to strengthen the security of WSNs within the context of IoT. Here are some contributions and schemes that can be utilized:

a. Secure clustering and data aggregation routing strategies

Building a safe framework for clustering and data aggregation routing algorithms is the primary focus of this study. This framework will reduce communication costs and increase data transmission security in wireless sensor networks. The proposed framework uses clustering algorithms and data aggregation routing strategies to reduce communication costs and improve the security of data transmission. This framework will also provide secure data transmission, aggregation, and routing. The contributions of this research work are as follows:

- It presents a framework for secure clustering and data aggregation routing algorithms to improve data transmission security while decreasing communication costs in wireless sensor networks.
- It utilizes clustering algorithms and data aggregation routing strategies to reduce communication costs and increase the security of data transmission.
- It provides secure data transmission, aggregation, and data aggregation and routing.
- It evaluates the performance of the proposed framework in terms of throughput, latency, and communication cost.
- It provides a comparison of the proposed framework with existing methods.

In addition, a model of a machine Learning-based Hybrid Neural Network is proposed to boost the Classification Accuracy; this model makes use of hybrid optimization methods to determine the optimal weights and maximize the network's performance. Finally, the proposed model's accuracy, precision, recall, F-score, training duration, etc., are compared to state-of-the-art, and the results are superior.

b. Secure localization based on blockchain technique in IoT-WSNs

The rapid development of the Internet of Things (IoT) and wireless sensor networks (WSNs) has enabled many applications, such as smart cities, intelligent healthcare, and connected cars. However, the security and privacy of IoT-based WSNs remain an unsolved challenge. Secure localization is an important security requirement for many IoT-based WSNs applications. Due to computational complexity and scalability issues, traditional secure localization techniques are difficult to deploy in large-scale WSNs. This research proposes a secure localization technique based on blockchain and federated learning for IoT-based WSNs. Secure localization is a key component of any WSN, providing the information needed for an accurate positioning system. However, traditional localization techniques are vulnerable to malicious attacks, as they rely on a centralized system to manage and process data. We proposed a secure localization technique based on distributed trust models and decentralized algorithms to guarantee privacy and security. The proposed secure localization technique uses blockchain and federated learning for IoT-based WSNs. Federated learning is used to train and update the blockchain-based secure localization models in a distributed manner. The proposed secure localization technique can provide secure localization in large-scale WSNs and protect data privacy and security using blockchain and federated learning. The proposed secure localization technique can be used in many applications, such as smart cities, intelligent healthcare, and connected cars.

1. Combine blockchain technology with federated learning, which will enable the creation of a secure, distributed, and efficient localization system.
2. The technique will also incorporate privacy-preserving mechanisms to protect user data.
3. The proposed system will be tested with real-world datasets to evaluate its performance and security.
4. The results of this research will provide useful insights for developing secure and privacy-preserving localization techniques for WSNs.
5. Establish encrypted paths and data transmissions in a wireless sensor network based on the hierarchical Internet of Things to deliver services safely.
6. It successfully identifies and pinpoints malicious nodes' location using a beacon node and the blockchain technique for computing the identities of previously unknown nodes.

c. Secure localization techniques

Wireless sensor networks (WSNs) have been widely deployed in various fields of application. In a WSN, sensor nodes are vulnerable to attacks like node replication, message spoofing, etc. Traditional security mechanisms are not enough to provide secure functioning in the WSNs. Therefore, it is necessary to develop new methods for localizing and detecting multiple attacks. Range-based and range-free localization techniques are two of the most widely used localization techniques in WSNs. To better localize and identify numerous attacks in WSNs, this study suggests employing Artificial Neural Networks (ANNs). ANNs are a class of machine learning algorithms inspired by the human brain's biological neural networks. They can learn complex relationships between data and can be used to solve complex problems. This research uses ANNs to localize and detect multiple attacks in WSNs. The ANNs are trained using data collected from the WSNs. The trained ANNs are then used to detect and localize multiple attacks in the WSNs. This research will provide valuable insights into using ANNs for localization and detection of multiple attacks in WSNs. The results of this study can be used to develop more secure and reliable WSNs.

The thesis, "Enhancing Security in Wireless Sensor Networks Integrated to IoT Using Different Schemes," encompasses seven chapters, conclusions, future directions, and a bibliography.
The thesis entitled as will focus on the security of wireless sensor networks integrated into the Internet of Things (IoT) using different schemes. The research will be conducted in four distinct stages, examining the following topics:

- We investigate the security vulnerabilities of wireless sensor networks and how they pose a risk to the IoT-WSNS.
- We can assess performance analysis of the different security schemes and technologies regarding attack detection rate, error rate, scalability, and reliability.
- We are developing comprehensive security frameworks and techniques for integrating wireless sensor networks and the IoT.

The research will also evaluate the effectiveness of the security framework in protecting wireless sensor networks and the IoT from malicious attacks and unauthorized access. To conclude, the thesis statement will make recommendations
The thesis work is organized as follows:

Chapter 1: Introduction

This chapter will cover the motivation and purpose of the outlined research topic concerning security in wireless sensor networks integrated into the Internet of Things. It will also contain the main idea for the development of the thesis. This section also encompasses the framework and thesis highlights of and contributions and application scenarios of the research work.

Chapter 2: Review related works based on various security techniques for IoT-WSNs.

This chapter encompasses a detailed study of the related work on security in wireless sensor networks integrated into the Internet of Things. The literature review of the recent works is conducted based on techniques for enchaining security. Some of the approaches are:

- Designing and developing a framework for advanced intrusion detection techniques in

WSNs for attack detection, prevention, and response

- Secure location techniques based on range-based and range-free strategy
- Secure clustering and data aggregation approaches in WSNs.
- Secure communication and routing strategies and techniques in WSNs.

Chapter 3: Design of secure intrusion detection systems

This chapter highlights developing and designing a framework for secure intrusion detection systems in wireless sensor networks. Machine learning techniques are utilized using benchmark datasets for attack detection and classification.

Chapter 4: Design and developing of secure localization framework

This chapter will highlight the discussion of the methodology adopted to enhance the localization accuracy of sensor nodes and minimizes the localization error of the unknown nodes. This helps detect malicious nodes in WSNs by computing the position and location of the unknown nodes.

Chapter 5: Blockchain-enabled secure localization based on federated learning

This chapter aims to use the distributed and public nature of blockchain technology and the private data protection features of federated learning to overcome security obstacles. Federated learning allows multiple devices or entities to collaboratively train a machine learning model without sharing their raw data, thereby preserving privacy.

Chapter 6: Secure clustering, data aggregation, and routing strategies

This chapter will discuss the methodology adopted to develop secure data aggregation and clustering approaches based on various routing strategies in WSNs. This helps detect malicious nodes in WSNs by secure clustering and trust evaluation techniques.

Chapter 7: Conclusions & Future Work

This section will briefly summarize all the research works, including network planning, simulation scenarios, and results. This section also summarizes the implementation techniques for localization and classification attacks in WSNs. The frameworks and procedural approaches for the proposed objectives are evaluated in terms of various performance metrics and research outputs. The undiscovered future works are also highlighted in this section.

Chapter 2

2 LITERATURE REVIEW

2.1 Introduction

This section studies various literature surveys based on intrusion detection systems, machine learning, secure localization, secure routing and communication strategies, and data aggregation techniques to enhance security and network lifetime. Security is essential in WSNs due to the random nature of network configuration that exposes them to several attacks. Therefore, different algorithms and protocols are examined for security and data aggregation in WSNs. It is an efficient method for saving traffic and network performance in WSNs. Surveying the existing literature in these areas, gaining insights into the state-of-the-art techniques, identifying research gaps, and determining promising directions for further exploration are important. The surveys help understand the effectiveness and limitations of different algorithms and protocols, paving the way for developing novel and robust solutions for enhancing security and network lifetime in WSNs integrated into the IoT.

2.2 Intrusion Detection Systems

L. Han et al.[6] examined a technique for reducing energy consumption and ensuring high efficiency using intrusion detection model theory, an autoregressive model. This method improves energy consumption and defence strategy. A. H. Farooqi et al. [7] proposed secured wireless sensor networks using a novel framework intrusion detection scheme against routing attacks. It works online and offline modes. S. Pundir et al. [1] reviewed security in WSN using the thread IoT communication model. They also discussed WSN security requirements against various attacks. They also critically reviewed intrusion detection protocols. A. Keramatpour et al. [32] studied the IDS in wireless sensor networks for data collection sensors deployment. They also provided real-world environments and actual conditions using simulations. U. Ghugar et al. [5] examined the detection of attacks in different WSN layers using protocol layer IDSs for secured WSNs. They also presented the different layers and protocols for secured WSNs. They also discussed the physical layer, media access control, and network layer for securing the network using the protocols. R. Singh et al. [29] Presented a cutting-edge Advanced Hybrid Intrusion Detection System (AHIDS) that can spot attacks on WSNs on its own. AHIDS employs a cluster-based architecture with an improved LEACH protocol to lessen the load on the power supply of the sensor nodes. The AHIDS employs the Multilayer Perceptron Neural Network and fuzzy rule sets for anomaly identification and misuse detection. Combining the outputs of a Feed Forward Neural Network and a Backpropagation Neural Network, we can identify and classify potential threats (i.e., Sybil attack, wormhole attack, and Hello flood attack). An Advanced Sybil Attack Detection Algorithm is created for Sybil attack detection, while a Wormhole Resistant Hybrid Technique is created for wormhole attack detection. The detection rate for a Sybil assault is 99.40%; for a Hello flood attack, it's 98.20%; for a wormhole attack, it's 99.20%. Ö. Cepheli et al. [33] examined a hybrid intrusion detection technique based on parallel combinations of detection methodology using flexible and tunable parameters. The hybrid anomaly and signature detection technique collect the output for the decision of the attack detection. The detector has parameters guided by a central node, a hybrid intrusion detection engine. Hybrid intrusion detection systems enhance DDoS attack detection accuracy. Figure 2. 1 shows signature-based and anomaly-detection block diagrams for attack detection as normal traffic and DDoS attacks.

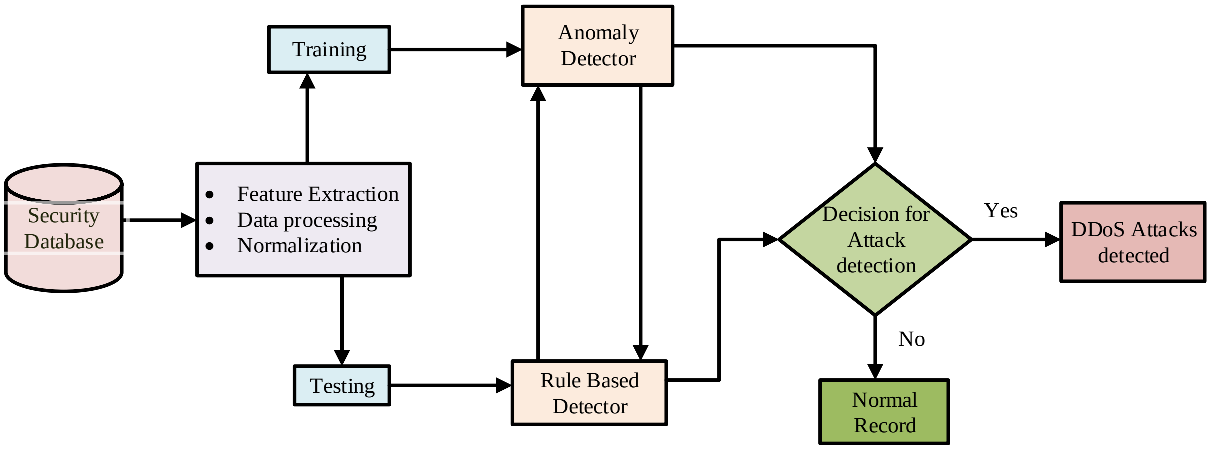


Figure 2. 1. Hybrid intrusion detection model with anomaly detector and rule-based detector.

The detection process starts with analyzing network traffic and extracting the building activity model features. The dataset DARPA 2000 evaluates the intrusion detection system with the DDoS attacks and normal network traffic records. The model is trained and tested with normal network traffic data to detect DDoS attacks. Hybrid detection uses anomaly and signature detectors for attack detection. The anomaly detector detects the feature extraction's normal and abnormal traffic data. The signature-based detector uses predefined sets to extract the traffic data's features. The detector in the signature detection scheme controls the set of rules applied to the system. The hybrid detection engine evaluates the security performance of the system.

The hybrid detection engine calculates the final decision using the probability density function. Advanced HIDS combines anomaly and misuse detection methods to detect routing attacks in WSNs. AHIDS uses a machine learning strategy containing two essential components, as shown in Figure 2.2. Anomaly detection uses blocks to identify data packets as normal or abnormal. The anomaly detection models utilize anomalous packets for the detection of malicious nodes. The misuse detection block recognizes various types of attacks using abnormal data packets. The detection of malicious nodes using AHIDS based on the fuzzy rule has three steps. The steps are:

- It measures the transmission of data packet history in WSNs through base nodes.
- It selects the feature set and looking the key elements for packet classification.
- Anomaly intrusion detection techniques are established based on data packet resolution.

The fuzzy-based intrusion detection technique uses MPNN, which consists of BPNN and FFNN for anomaly and misuse detection, as shown in Figure 2.2. It is applied for the highest detection rate using supervising learning technique. The fuzzy base AHIDS with FFNN and BPNN achieves greater attack detection accuracy using massive clustered training. The multilayer perception is utilized for estimating the error rate, e_i, using the formula (2.1).

$$e_i = d_i - a_i \tag{2.1}$$

Where d_i represents the preferred output and a_i is the true output obtained from MPNN.

The decision-making module uses anomaly and misuse detection schemes to detect attacks.

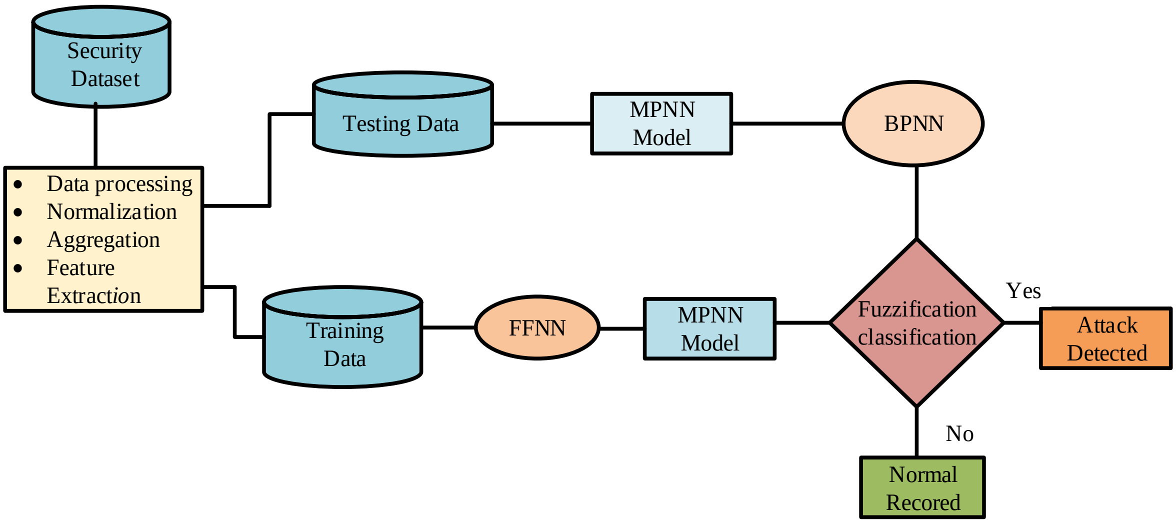


Figure 2.2. Block diagram for advanced and hybrid intrusion detection system.

The decision-making process aggregates the outputs using the fuzzy rule. The fuzzification process assigns the input parameters for FFNN and moves them to the fuzzy inference system.

The fuzzy rules are applied to determine the attack type producing defuzzification output mild fuzziness, high fuzziness, and low fuzziness for the wormhole, flooding, and Sybil attacks, respectively.

The MPNN model consists of BPNN and FFNN and is applied to evaluate the detection accuracy of the various class attacks in WSNs. The MPNN utilizes BPNN and FFNN techniques for IHIDS to manage huge datasets and the system's stability. FFNN detects the new type of attacks, and BPNN clusters the mysterious attacks for MPNN supervised learning. The membership vector applied for the fuzziness F(V) is given by equation (2.2):

$$F(V) = -\frac{1}{n}\sum_{k=1}^{n}(\mu_i \log\mu_i)+(1-\mu_i)\log(1-\mu_i) \tag{2.2}$$

Where $V=\{\mu_1, \mu_2,..., \mu_n\}$ is a set of fuzzy, the fuzziness values are categorized into high, low, and mid fuzziness groups with training and testing samples.

L. Gandhimathi and G. Murugaboopathi [34] researched flow-based and cross-layer hybrid intrusion detection to detect anomaly traffic and narrow features for possible attacks. Malicious nodes are classified using flow-based IDS during the first phase. The packets are verified and detected using cross-layer feature analysis. The flow-based IDS monitors the traffic on the network using network information and classifies the data as normal or malicious, as shown below in Figure 2. 3. By keeping tabs on network activity and keeping track of many parameters, flow-based anomaly detection can exhibit a regular profile of behaviour. The flow-based intrusion detection scheme identifies the parameters, including network connection, hosts, users, and applications. The anomaly detection technique uses the recorded network traffic activities during the training and detection phases. There is a connection between the cross-layer characteristics and the packet-based IDS. The network layer extracts the routing information when determining the forwarding path nodes. Attack detection relies on cross-layer correlation and comparison of MAC and network properties. To lower the false positive rate and improve detection accuracy in WSNs, the hybrid IDS combines the flow based and the packet based.

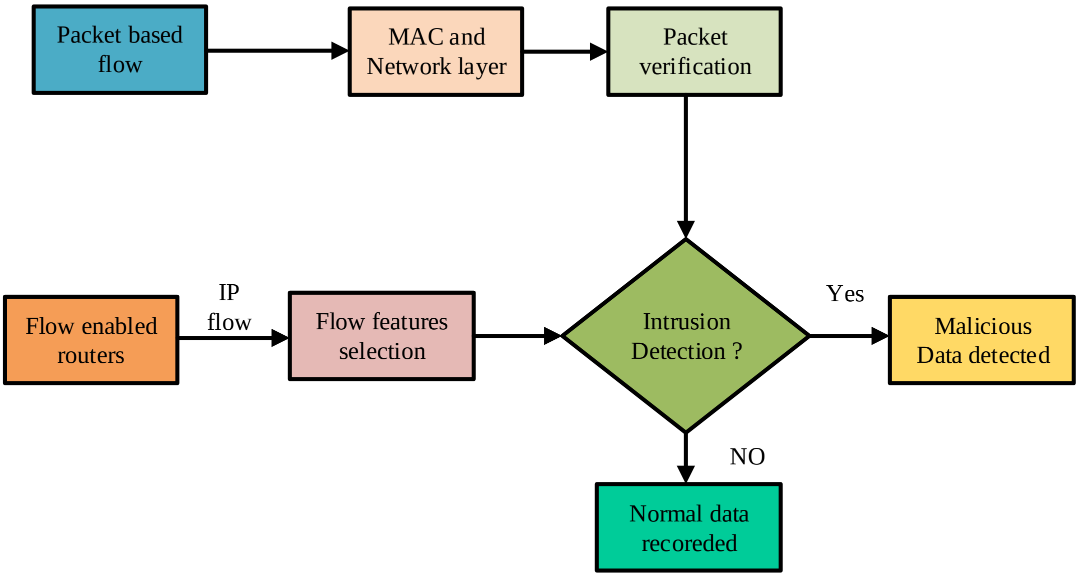


Figure 2. 3. Block diagram of an intrusion detection system using the flow-based technique.

DoS and sinkhole attacks are easy to spot when many flow-based and layer IDS correlate. Cross-layer IDS improves detection accuracy by correlating the protocol layer, while flow-based IDS uses the flow record to identify attacks.

L. Moulad et al. [35] proposed a hierarchical hybrid intrusion detection technique using a support vector machine and anomaly detection-based clustering and specification techniques in WSNs to detect and classify DoS attacks. The specification, signature, and anomaly detection techniques are depicted as shown in Figure 2. 4. The specification-based technique used the deviation technique for identifying normal behaviours from malicious data. They are described as behavioural and manually defined specifications for monitoring and taking action against the attack. The specification approach adopts a statistical model for detecting anomalies from normal behaviour, including intrusion attacks in the network. This technique offers better flexibility concerning parametric learning using well-defined threshold settings. The signature-based techniques have a predefined set of rules based on the known security attacks, and the system is designed based on the knowledge of attacks for pattern detection. In contrast, the anomaly detection technique utilized a threshold baseline for comparing the behaviours of malicious nodes with the normal nodes for establishing automated training using support vector machines. Support vector machines are supervised machine learning methods for detecting and classifying network behaviour and acting as binary classification techniques.

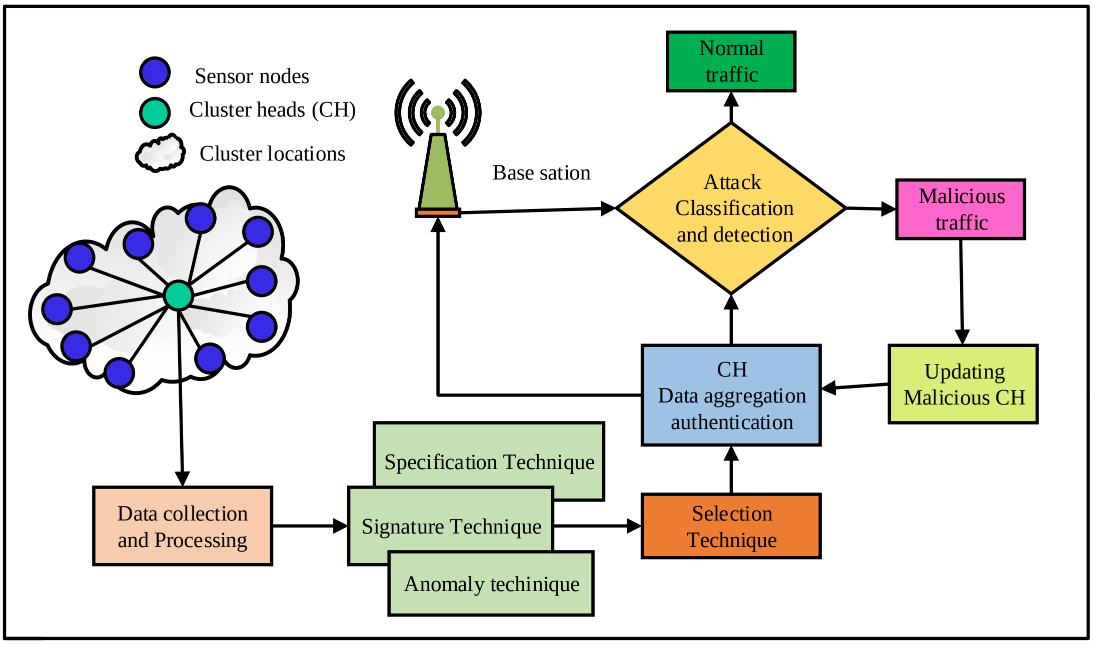


Figure 2. 4. Hierarchical hybrid intrusion detection technique in wireless sensor networks using three phases of attack detection.

The hierarchical hybrid intrusion detection system (HIDS) used a clustering technique for data aggregating and increasing the network lifetime. The sensor nodes collect information from the environment, send it to the cluster head (CH), and then forward it to the base station in the network. A. Abduvaliyev et al. [36] proposed hybrid intrusion detection techniques to improve network security and energy performance by increasing computational costs and reducing communication overhead using clustering and aggregation techniques in WSNs, as shown in Figure 2. 5.

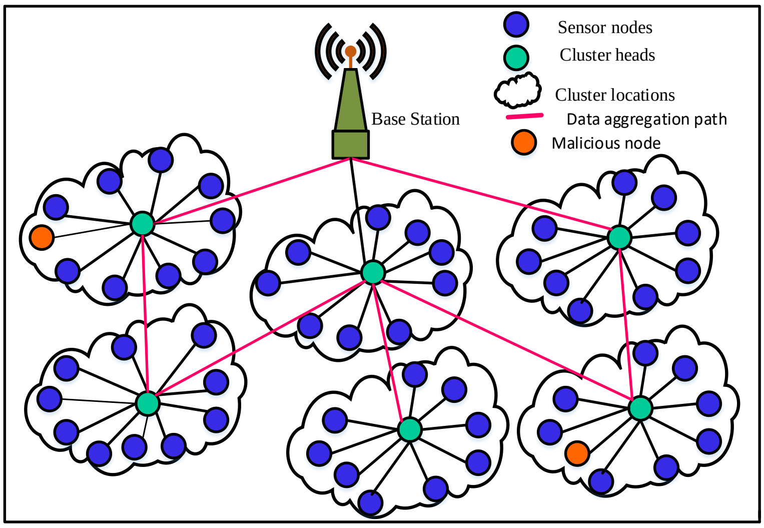


Figure 2. 5. Illustration of data aggregation and clustering in WSNs.

The cluster head in the network manages the operation sensor nodes in the cluster and aggregates the data to reduce packet overhead and energy consumption. This energy-efficient hybrid intrusion detection system utilized misused and anomaly detection techniques to improve detection accuracy and rate. Misuse detection uses known attack patterns, whereas anomaly detection uses automated training behaviours.

R. M. Swarna Priya et al. [37] proposed deep neural network models helpful in improving and reducing dimensions, detection, and classification accuracy using hybrid principal component analysis and grey wolf optimization (PCA-GWO) metaheuristics for designing intrusion detection systems using benchmark datasets in wireless sensor networks for medical applications. The principal component analysis is a technique for reducing and transforming the dimensions of a large amount of data without losing the major components of the dataset. The processed data is fed into training and testing using a fully connected deep neural network using autoencoders with trainable parameters. P. R. Kanna and P. Santhi [38] examined hybrid deep learning models for designing effective hybrid intrusion detection systems using MapReduce based on black window optimized long-term convolutional memory using feature selection techniques. The artificial bee colony technique is utilized for feature selection, followed by the hybrid deep learning classification technique using benchmark datasets for intrusion detection systems.

Applying a K-medoid clustering strategy on a synthetic dataset, Misdetection is used to detect blackhole attacks [27]. The hybrid anomaly detection method uses the K-medoid individualized clustering approach for diversion and blackhole assaults. A synthetic dataset was produced by establishing network parameters, and threshold values were derived to identify the outliers. The NS-2 network simulator and the R statistical programming environment were used for the experiments. The proposed algorithm achieves an accurate detection of hybrid anomalies. This technique applies to detecting hybrid anomalies in a wireless setting, such as blackhole and misdirection assaults. SDN-based Hybrid clone node detection technique detects the cloned node using proactive and verification processes based on software-defined networking in WSNs, maintaining and enhancing the quality of service [30]. Quality of service (QoS) limitations can be better maintained and enhanced using this SDN-based method in WSN. A clone node in a wireless network can be identified using the hybrid clone node detection (HCND) technique. The goal is to effectively detect clones so that cloning attacks can be stopped before they even begin. Clones can be uncovered regionally and globally through a low-cost identity verification process. This technique helps secure a WSN by preventing cloned nodes from impersonating legitimate ones through a superimposed SDIS junction code. The most secure route can be chosen if many copies of a node's identification are stored in different locations. All network nodes must use the layered approach to retrieve data. Getting rid of the clones hosting the cluster of attacks is the best way to defend against them. Multiple metrics, including sensitivity, specificity, false positive/negative ratio, precision, recall, and detection, were analyzed in the simulation results.

Anomaly-based and signature-based detection enhances overall accuracy using DARPA 2000 public dataset as a benchmark [33]. A secured intrusion detection system (IDS) hybrid approach provides a unique security technique for preventing and detecting attacks. The scheme realizes data integrity, authentication, and energy minimization [39]. The Adhoc on-demand dynamic vector scheme is useful for detecting routing attacks using hybrid security techniques in wireless sensor networks. Watchdog and Delphi hybrid techniques provide computed probability factors for detecting wormhole attacks in wireless sensor networks [40]. The scheme effectively detects and classifies Grayhole and blackhole attacks by deploying hybrid security mechanisms. It also provides secure data transmission and prevention of multiple attacks with strong encryption and positive performance. Hybrid multi-tiered IDS using machine learning is also practical for detecting internal and external cyber-attacks targeting vehicular networks using the CICIDS2017 dataset [41]. They are also used for reducing energy consumption and malicious anomaly nodes using lightweight IDS and signatures based on clustering using the support vector machine technique [42]. Hybrid IDS is also used for cloning attacks using the Clone detection technique to detect and verify cloning attacks in WSNs [43]. Hybrid IDS techniques utilize a feature selection approach for detecting and classifying IoT-based security attacks for healthcare applications using hybrid DT-GA with minimum and optimal cost [44].

2.3 Machine Learning Techniques

M. Rabbani et al. [45] presented a combination of machine learning and traditional security technique that provides adequate protection and intrusion detection system. The new scheme justifies and provides a mathematically secure platform for all nodes. This technique detects and recognizes the malicious behaviour of the attacks using the data pre-processing and recognition modules. The pre-processing data stage is designed for data preparation and extraction of features for modelling in machine learning techniques, as shown in Figure 2. 6. The recognition techniques are designed with training and prediction phases for an optimized probabilistic neural network using the UNSW-NB15 dataset containing malicious and normal data traffic networks.

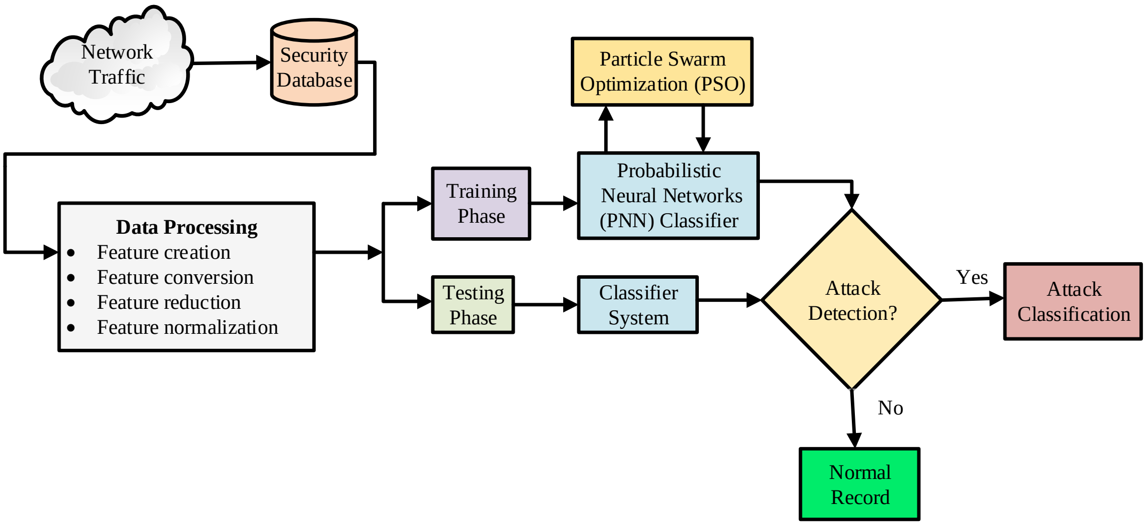


Figure 2. 6. Hybrid PSO-PNN technique for attack classification and detection.

The hybrid PSO-PNN scheme shown in Figure 2. 6 is used to build a self-optimized network. The PNN uses training and a feedforward approach in a faster way. The combination of PSO-PNN reduces misclassification errors using adaptive standard PNN design. The PNN finds the statistical properties and predicts the classification accuracy. Network traffic patterns' normal and malicious activities are classified as normal records and attacks. The PSO is an optimization technique to simulate behaviour patterns and increase the classification rate in the PNN system. The PSO makes the PNN structure into a self-adaptive network model using particle and swarm folds.

The features of the data network traffic are collected from the raw network packets using tools including Netmate, BRO-IDS, and Argus. The noisy features are removed to detect malicious attacks in WSNs effectively. Numeric values and symbolic variables represent the necessary features. The numeric and symbolic representations are normalized and transformed using the statistical characteristics given by equation (2.3):

$$Z_{\text{normalized}} = \frac{(Z\text{-}\min(Z))}{(\max(Z)\text{-}\min(Z))} \tag{2.3}$$

Where Z is the feature value, min (Z) is the minimum value, and max (Z) is the maximum value from the feature of the samples in the dataset.

S. M. Kumar [46] presented an optimized hybrid deep neural network using a feature section algorithm for improving the intrusion detection of attacks by combining long shirt term memory and convolutional neural network using benchmark datasets UNSW-NB15 and NSL-KDD. The selected features are processed into the convolutional neural network consisting of layers using distance measurement and the correlational coefficient for input packets. The distance between two data points (a_i, b_i) with x and y input data for arranging and selecting the features is given by equation (2.4) as shown below:

$$D(x,y) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{2.4}$$

A convolutional neural network was utilized to process the selected features. The outputs are fed into the long-term memory configuration and modified materials to enhance classification accuracy.

S. Mahajan et al. [47] examined statistical and non-statistical algorithms utilized as hybrid machine learning and deep learning techniques for network traffic analysis and classification in wireless sensor networks. The scheme is also used for attack detection and classification approaches using benchmark datasets. J. Al Faysal et al. [48] proposed machine learning techniques to detect attacks in IoT-based wireless sensor networks using benchmark datasets. Hybrid eXtreme Gradient Boosting and Random Forest (XGB-RF) successfully detected botnet attacks using feature selection and classification with various metrics. M. I. Alghamdi [49] created a novel optimizer technique using cascade forward neural network (PO-CFNN-) based IDS in the IoT setting. The primary goal of the PO-CFNN method is to identify intrusions in an IoT setting. The PO-CFNN method has three stages: preprocessing, classification, and parameter optimization. The networking information is first put through a preprocessing stage, where it is transformed into a more usable format.

F. Sadikin et al. [50] researched the combination of anomaly and rule-based machine learning techniques to detect attacks on ZigBee-based IoT systems using the hybrid intrusion detection system. This hybrid technique is implemented in the rule engine by re-running the testbed with the normal behaviour-producing attack scenarios. The rule-based intrusion detection scheme uses rules that humans drafted using their knowledge. The scheme creates anomaly detection rules for detecting attacks based on malicious behaviour. The various types of attacks can be detected by analyzing the datasets of attack scenarios using the legitimate node behaviour as the threshold. The various types of attacks, like flooding attacks, replay attacks, and TouchLink Inter-PAN attacks, can be detected based on the normal behaviour of the authorized node using signal strength pattern, frame counter, traffic rate, and packet frame format. The machine learning model has various features, including received signal strength, time, frame counter, and packet interval for the ZigBee frame format.

Hybrid Convolutional neural network (CNN) and long short-term memory network (LSTM) learning approach Extracts the network data traffic features using the hybrid IDS with the CICIDS2017 dataset for evaluation and achieves an overall accuracy of 99.50% for attack type [51]. The sink node was trained with a predictive classifier employing a Kalman filter (KF) and an extreme learning machine (ELM) hybrid classification technique. The scheme is evaluated using the detection of random WSN anomalies data with the normal and faulty datasets [52]. For binary classification, the Hybrid k-means and support vector machine are used for reduced training and testing times with promising classification accuracy of attack detection and classification in the network [53]. The Federated learning techniques are utilized for a privacy-friendly framework across multiple devices using benchmark datasets [54]. Deep learning-based hybrid neural network techniques are effective for detecting and classifying attacks. IoT enables networks to utilize the hybrid chicken swarm genetic algorithm method [55]. The strategy implemented hybrid optimization and deep-learning-centric intrusion detection systems to address these challenges in IoT-enabled smart cities. A trustworthy IDS cannot be obtained without first pre-processing the dataset. The Hybrid Chicken Swarm Genetic Algorithm (HCSGA) and the K-means Algorithm are then used for feature selection and grouping. At last, the normal and attack data are classified using the NSL-KDD benchmark dataset and fed into the Deep Learning-based Hybrid Neural Network (DLHNN) classifier, which is used to verify the model.

Hybrid machine learning techniques utilize sampling methods to provide better detection accuracy using feature selection analysis techniques [56]. To address the issue of positive and negative sample imbalance in the original dataset, we employ a hybrid sampling method that combines Adaptive Synthetic Sampling (ADASYN) and Repeated Edited nearest neighbours (RENN) for sample processing. By utilizing a mix of the Random Forest method and Pearson correlation analysis, we can overcome the issue of feature redundancy in the feature selection process. The spatial features are then extracted with the help of a convolutional neural network and then further extracted with the help of fusing Average pooling and Max pooling, assigning different weights to the features with the help of an attention mechanism, decreasing the overhead and increasing the model's performance. In addition, a Gated Recurrent Unit (GRU) is used to extract the long-distance dependent information features to accomplish efficient and thorough feature learning. We classify using a softmax function. It has been shown that hybrid deep learning methods can effectively identify malicious attacks when training and testing on performance benchmark datasets [57].

2.4 Anomaly Detection

M. Wazid and A. K. Das [28] proposed hybrid anomaly detection for multiple attacks, such as misdirection, wormhole, and blackhole attacks launched using the hybrid anomaly technique. Hybrid anomaly detection is a technique for intrusion detection using the K-means clustering approach. M. Wazid [58] examined hybrid anomaly detection techniques utilizing threshold values for detecting anomaly traffic data with a k-means clustering scheme. The anomaly techniques are evaluated using a dataset with the abnormal and average values of the parameters in the network. C. Umarani and S. Kannan [59] proposed hybrid anomaly detection techniques that are regarded as artificial immune systems in wireless sensor networks using hybrid tissue growing techniques to detect malicious traffic. The hybrid tissue growing technique combines networked and swarms tissue growing approaches for effective anomaly recognition using tissue structure for transmitting data packets.

C. Yin et al. [60] presented an anomaly detection technique that recognizes and separates normal and abnormal behaviours using the patterns of normally labelled behaviours. The data mining technique extracts valuable knowledge from substantial data records to recognize the patterns of malicious nodes. This can be classified as regression analysis, clustering analysis, outlier detection, and classification. The data mining application in anomaly detection improves the detection accuracy and efficiency of the network. It is essential for information security and provides all-around performance for intrusion detection. Anomaly detection can identify attacks, including spoofing, packet injection, flooding attacks, replay attacks, and other attack activities based on predefined rules of normal behaviour. Anomaly detection requires a machine-learning model with human effort and is error-prone.

2.5 Localization Techniques

An online consecutive distance-vector hop technique for node localization accuracy in WSNs was presented by S. Messous and H. Liouane [10]. They also talked about how the distance between anchor nodes can be optimized. For better security in WSN, S. Dong et al. [3] tested the distance-vector hop algorithm against Sybil assaults, finding that it provides effective node localization with high accuracy. Setting the beacon nodes at 50 in the simulation yields a 78%

reduction in average error localization thanks to the technique. Mobile WSNs need a localization algorithm, which L. Chelouah et al. [61] discussed. The flexibility of nodes in terms of coverage optimization, connectivity, and analysis was also demonstrated. Localization in WSNs was given by A. Hadir et al. [62] utilizing a powerful distance-vector hop algorithm. The typical hop distance and localization precision gained from this data are also discussed. H. Chen et al. [63] developed an intelligent, low-cost method of detecting and preventing DoS attacks. In addition, they classify DoS attacks using a unique dataset designed for WSN and talk about the results. Many methods for detecting Sybil nodes [64] were provided by S. T. Patel and N. H. Mistry [65]. The protocols employed by WSNs were also the subject of discussion and analysis. It was proposed by F. Y. Yavu et al. [66] to use a deep learning machine learning technique to identify IoT routing attacks. Coojia's IoT network simulator uses a thousand sensors to create realistic data on attacks. Sybil attack detection utilizing hybrid fuzzy and strong extreme learning machines was studied by V. Sujatha and E. A. M. Anita [67]. Further, they talked about real-time Testbeds that use ARM as the primary CPU in a LEACH setting with Zigbee transceivers. To enhance the accuracy of the node positions and decrease the localization error in WSNs, X. Qi et al. [68] investigated a localization technique based on MA-MDS. To ensure precise coordinate transformation, they employ the Prussian analysis algorithm. To uncover spoofing and Sybil attacks, P. Li et al. [69] introduced a localization trust valuation scheme. This scheme is obtained when a threshold property is applied to selecting localization performance, estimated distance, and transmission in WSNs. Using a glowworm swarm optimization strategy, L. Song et al. [70] suggested a chaotic hybrid mutation and chaotic inertial weight update mechanism. The technique also improves convergence and accuracy while avoiding premature convergence. Sybil attack detection for global mobile communication networks utilizing signed response authentication approaches was provided by M. Saud Khan and N. M. Khan [71]. The probabilistic methodology for evaluating Sybil attack detection performance was also described.

2.6 Blockchain-based Secure Localization

Blockchain methods can be used to track down and name malicious nodes in IoT-based wireless sensor networks. The literature review examines several related publications and techniques. Among the literary schemes are:

- Malicious nodes in networks of low-power sensor nodes can be safely pinpointed.
- Routing strategies for efficient use of resources and long network life

- Establishing credibility between nodes using authentication and encryption.
- A blockchain-based localization mechanism is used to improve WSNS's quality of service.
- Optimizing and securing data storage in IoT-WSNs: how to do it.
- Models that are both secure and efficient are needed for IoT-based WSNs of the future.
- Methods of machine learning and categorization that are collaboratively built and shared

H. Kim et al. [16] Investigations on blockchain methods for identifying rogue nodes have been conducted utilizing a trust management approach to improve the connections between beacon nodes. Selecting beacon nodes based on trust levels and having them provide data to the base station creates a trust evaluation model for finding and pinpointing malicious nodes in WSNs. Z. Abubaker et al. [19] presented a blockchain-based approach to detecting and localizing rogue nodes in IoT-based wireless sensor networks using federated random forest and support vector machine techniques. They also talked about identifying and eliminating hostile nodes in a network while providing safe service provisioning through feature evaluation and cascade encryption. The performance of malicious node identification and secure provisioning methods were evaluated using security and performance criteria such as accuracy, node honesty, and end-to-end packet transmission delay. C. Ming et al. [20] proposed a blockchain-based authenticated group key agreement mechanism for IoT devices. A novel idea known as the device manager is proposed in the proposed protocol to facilitate interactions between IoT devices and blockchain networks. The security study shows that the suggested protocol is safe even after being attacked in several ways. The simulation findings show that the protocol operations' time requirements are reasonable and suitable for IoT environments. R. Kumar et al. [72] presented a new distributed IDS based on fog computing to detect DDoS attacks against mining pools in blockchain-enabled IoT networks. A gradient tree boosting system (XGBoost) and Random Forest (RF) are trained on fog nodes, and their results are compared. The suggested model is tested on a real-world IoT dataset, including the most recent attacks in a blockchain-enabled IoT network. The DDoS protection system was integrated with a mining pool in a blockchain-enabled Internet of Things network. The proposed distributed detection system is powered by three separate processes. In the outset, there is the traffic processing engine, which uses fog nodes to preprocess network data by standardizing its properties with the aid of the aforementioned Standard Scaler.

R. Goyat et al. [73] tested a blockchain-based range-free localization method in wireless sensor networks against a malicious node in a 2D setting for its security and novelty. In addition, they talked about how to put blockchain technology into practice by employing the trust value of beacon nodes in the mining process of blocks to create a geolocation procedure for unknown nodes based on their neighbour node list, mobility, residual energy, and repudiation value. Unknown nodes' positions are estimated using beacon nodes with verified trust values created by the blockchain to determine their exact locations. The malicious nodes spread misleading localization information by hijacking the beacon and sending inaccurate energy data. S. Awan et al. [74] proposed blockchain-based detection and encryption-and-trust evaluation approach. To identify malicious nodes and prevent them from wreaking havoc on the network by, for example, transmitting false routing and energy data or compromising the aggregating node and thereby increasing resource consumption and packet loss, we authenticate sensor nodes and aggregator nodes in a private blockchain and a public blockchain, respectively. After authenticating the sensor nodes, the detection rate was used to calculate the trust values and residual energy of the nodes to remove the malicious nodes from the secure routing in the network.

H. H. Pajooh et al. [75] proposed a multi-layered blockchain security architecture to ensure a safe Internet of Things. Multi-hop cellular networks are utilized by IoT devices, and the unidentified sensor nodes within the network are clustered using a self-clustering hybrid evolutionary strategy that merges simulated annealing and genetic algorithm methodologies. The cluster head performs authentication and authorization locally so that it can communicate efficiently with the base station using the increased throughput and decreased network latency provided by a private lightweight, balanced blockchain. S. Otoum et al. [76] introduced a flexible framework for evaluating network trust that uses blockchain technology and federated learning, namely reinforcement support vector federated learning. Models were developed for communicating with fog and building a global model, and reinforcement learning happened at the end devices. They used simulation approaches to test and validate the model's assumptions and predictions, using measures like trust value, energy usage, and network longevity as performance indicators. The model's detection rate for rogue nodes was 96%, and its accuracy for pinpointing their location was 93%, thanks to its use of federated learning. She et al. [77] Built a blockchain data structure in three-dimensional space using a localization mechanism. They proposed its use as a framework for identifying rogue nodes in IoT-based wireless sensor networks. A quadrilateral measurement uses intelligent contact and wireless sensor networks to pinpoint rogue nodes. Validating and recording sensor nodes in the blockchain network helps do away with and prevent the loss of routing data. TK. Fan et al. [78] Suggested a blockchain-based, distributed-topology strategy for securing IoT systems that can identify and analyze threats in real time utilizing various time stamps. The method uses an immutable blockchain to keep track of timestamps and broadcast them for use in attack detection, with an adaptive topology to cut down on unnecessary communication. By utilizing a flexible topology and a distributed block structure, the technique prevents data from being accumulated in a single location.

O. Friha et al. [79] Introduced a federated learning-based IDS for agricultural data protection and security, leading to an enhanced threat detection model in the Internet of Things. The strategy used three distinct types of neural networks: convolutional, recurrent, and deep to detect attacks and categorize key performance characteristics from standard datasets. Accuracy, F1-Score, and Recall were also included as performance metrics for the federated deep learning model used in the multiclass categorization. N. Javaid [80] examined safe and effective communication between sensors and sink nodes in IoT-based WSNs free from the threat of hole attacks, and a new routing protocol based on the Dijkstra algorithm was studied and developed. With the blockchain method used by the network to identify malicious nodes, the system also enabled transparency for transactions conducted by the sensor nodes. Proof-of-authority consensus was used to add transactions to blocks, verifying the trust architecture regarding energy and detection techniques. D. Wu and N. Ansari [81] presented to identify potentially harmful nodes in the IoT infrastructure of industrial applications, a blockchain-based security and trust evaluation process has been developed. The blockchain can be used to store and track an access control list securely. To further increase the likelihood of identifying and authenticating rogue nodes in the network, they devised a voting approach incorporating trust evaluation into the access control list. A. U. Khan et al. [82] described how a hierarchical, low-power, energy-temperature-degree-adaptive routing protocol was implemented. The protocol's reduced energy requirements for nodes during data transmission contribute to the network's durability. The cluster heads (CHs) used to perform routing in the proposed protocol are chosen based on their degree, temperature, and energy. Additionally, the blockchain is utilised to remove the potential for a lone point of failure. Data exchanges between CHs and BSs are recorded and tracked using a blockchain, which requires several nodes for operation. All transactions on the blockchain use a safe hashing algorithm with 256 bits of computing power

(SHA-256). Finally, the routing process is protected from malicious nodes with the use of real-time message content validation (RMCV) in the ETD-LEACH protocol.

R. Goyat et al. [83] addressed the security concerns of IoT-WSNs and the vulnerabilities of trusted localization architecture attacks. Specifically, it looks at how a blockchain is made and how trust is assessed. Several trust measures are used to determine a beacon node's trustworthiness, and the relative importance of these metrics is adjusted on the fly during localization. This is why, during the mining process, only the most trustworthy beacon nodes are picked. This two-step procedure ensures that the blockchain is always accurate and that the trust values of all beacon nodes are accurate. D. Minoli and B. Occhiogrosso [84] Protection method in the framework of a defences-in-depth/Castle strategy based on blockchain mechanisms (BCMs) that protect a wide variety of IoT-oriented applications by becoming a piece of a security mosaic; proposed and implemented. A blockchain is a decentralized digital ledger that stores transactions and other data types in a permanent, unchangeable format that cannot be tampered with by any third party. By extension, these communications become available to everyone on the network. Information is stored and made available in a public ledger that all nodes in the system update simultaneously. In some cases, a fully blockchain-secured network may be impractical for Internet of Things uses; nodes with adequate capacities may be required to support the important peer-to-peer functionality in critical or institutional applications like smart grids, ITSs, e-health, insurance, and banking. However, these nodes don't always exist in the IoT. Z. Ma et al. [85] display a secure data-sharing mechanism for IoV information based on the blockchain with ITS as an illustration. With the help of smart contracts, it is possible to implement functions like instantaneous registration and authentication and a secure method of sharing IoV data. Confidentiality is maintained by homomorphic encryption and zero-knowledge proof processing performed by the smart contract and stored as ciphertext on the distributed ledger. To ensure the integrity of the network ledger, we employ a PBFT consensus mechanism and a Merkle tree-based block that cannot be altered to record all processes using IoV data. Under privacy and security, the IoV Chain technique keeps IoV data accessible and unaltered, making it possible to track it down if necessary.

X. Yang et al. [86] presented a blockchain-based ensemble anomaly detection (BCEAD) system, which saves the model of a common anomaly detection method on the distributed ledger to solve the problem of centralized anomaly detection in WSNs. The system was modified repeatedly after an appropriate block structure and consensus technique were established to improve detection and classification precision. The blockchain ensures a secure network environment, making the detection method impervious to system-level attacks. The results show that BCEAD outperforms rival schemes regarding performance, cost, and other characteristics. The sink layer is responsible for anomaly detection across the whole network. This technique for detecting isolation forests is stored on a distributed ledger (tangle). We also contrast experimental data to demonstrate that BCEAD outperforms prior techniques regarding detection efficiency. M. Sarhan et al.[87] introduced a blockchain-based hierarchical federated learning architecture to guarantee the confidentiality of collaborative IoT intrusion detection. The proposed machine learning-based intrusion detection system employs a federated learning architecture with a hierarchical structure to safeguard training and operational data. There may be various advantages to using Machine Learning (ML) skills to defend Internet of Things (IoT) systems from assaults. The existing framework proposals do not currently consider data privacy, safe architectures, and scalable deployments of Internet of Things ecosystems. All company transactions and activities will be managed via a secure distributed ledger, with promises enforced by smart contracts.

S. J. Hsiao and W. T. Sung [88] proposed a solution that was developed using blockchain technology to increase the security of data transmissions in wireless sensor networks (WSNs). Data transmission and distributed ledger technology (blockchain) form the backbone of a secure WSN in this research. Modern wireless networks are built on the Internet of Things concepts, and they use blockchain technology to guarantee the authenticity of every data in transit. In this study design, sensor data is recorded in a blockchain, a distributed ledger system. The reliability of the wireless sensing network architecture is enhanced by the proposed system's collection and analysis of sensor data. Cryptographic processing and the decryption of public keys are used in every blockchain to create a link between individual blocks. Moreover, in a brand-new blockchain, the preceding and following link sequences must be authenticated before any data may be communicated. S. Abbas et al. [89] introduced a minimal blockchain-based authentication system for storing the credentials of common sensors. As IoT nodes only have so much battery life, it's necessary to implement lightweight authentication by maintaining minimal credentials in the distributed ledger. Additionally, in an IoT network, the route calculation and on-demand routing are performed by a genetic algorithm-enabled software-defined network controller to reduce the energy consumption of the nodes. We also provide a route validity technique to ensure the proposed route is safe and free of any potentially dangerous nodes. E. H. Abualsauod [90] proposed study on unmanned aerial vehicle (UAV) IoT frameworks and analyze it to identify solutions to difficulties with the frameworks' overall security and privacy. Several security and reliability challenges exist in UAV-enabled IoT applications; however, the study provides an optimal solution for these problems by integrating various blockchain technologies. Results are compared and contrasted across several criteria, including processing speed, latency, accuracy, and overall system usefulness.

2.7 Secure Clustering, Data Aggregation, and Routing

This section studies various literature surveys based on secure clustering, data aggregation, and routing techniques to enhance security and the network's lifetime. Security is essential in WSNs due to the random nature of network configuration that exposes them to several attacks. Therefore, different algorithms and protocols are examined for security and data aggregation in WSNs. It is an efficient method for saving traffic and network performance in WSNs.

B. Bhushan and G. Sahoo [91] explored the security threats and vulnerabilities at different protocol layers. They also discussed data aggregation and routing protocols for energy-efficient WSNs. S. Gomathi and C. Gopala Krishnan [92] Proposed a method for detecting malicious attacks using secure data aggregation protocol in clustering and tree topology. In this scheme, aggregation of the nodes improves the network's trust and average detection accuracy. W. Min et al. [3] examined the security aggregation approach using a renewable hash chain for efficient data security. This technique provides data confidentiality, authentication, and integrity for WSNs. M. Kaur and A. Munjal [93] provide information summaries based on energy consumption, delay, network lifetime, and cost measures. A. A. A. Jasim et al. [94] Presented a protocol for addressing wireless sensor network security and energy issues. They deployed a protocol for extending safe data aggregation access control and authentication. X. Qi et al. [95] studied elliptic curves using asymmetric key encryption for security solutions. They also discussed hop verification using MAC protocol. The method optimizes the key and uses homomorphism to end encryption. K. P. Uvarajan and C. Gowri Shankar [96] examined a technique for enhancing global aggregation in WSNs. The scheme also enhances trust evaluation using energy and correlative divergence. A. Razaque and S. S. Rizvi [97] designed the secure data combination technique using the authentication and access control protocol. They also discussed the detection and challenges of routing attacks using the cryptography approach. The SDAACA protocol consists of authentication and data fragmentation algorithms for secure WSNs. X. Liu et al. [98] reviewed secure data aggregation techniques in WSNs. They also discussed the security strategy and topology of the network. Z. Zhang et al. [99] Proposed a method for heterogeneous WSNs using topology optimized based on local tree reconstruction. The nodes are distributed in different layers with different energy layers. The aggregation scheme prolonged the life and energy utilization of the network. C. Bekara et al. [100] examined a secure clustering and data aggregation in WSNs. The aggregation of sensors data produces small-sized output so that attackers are easily detected. The scheme has low overhead data transmission. R. Maivizhi and P. Yogesh [101] proposed a Q-learning-based novel adaptive routing algorithm for internal data aggregation in WSNs. The scheme applies reinforcement learning for building a routing structure using minimal information for optimal aggregation ratio. M. Mathapati et al. [102] developed a robust and secure data aggregation framework in WSNs. They also discussed energy efficiency, delay, and computational time using a flexible, lightweight encryption system in the network. M. Naghibi and H. Barati [103] presented a secure data aggregation technique combining the star and tree structures WSNs security by dividing into four parts creating the star topology. They also studied lightweight symmetric encryption for securing the data in the proposed hybrid structure data aggregation.

Several types of research are conducted on hybrid security techniques in wireless sensor networks. The hybrid techniques for detecting and classifying attacks are hybrid secure data aggregation, hybrid intrusion detection system, hybrid anomaly detection…, etc. They are a combination of two methods and are effective for WSN security. S. Gopikrishnan and P. Priakanth [104] proposed a hybrid secure data processing to provide energy-efficient, highly secure data aggregation. The scheme performs the private key generation and encryption at the leaf node. The method also reduces the computation and communication overhead of the sensor nodes. R. Singh et al. [29] proposed an advanced hybrid IDS to detect attacks in WSNs. The scheme used clustered-based architecture to reduce energy consumption by the sensors using the leach protocol. The technique uses a fuzzy rule with a multilayer layer perception neural network to use anomaly detection and misuse detection. M. Naghibi and H. Barati [103] Presented a secure hybrid technique using the combining star and tree topologies based on data aggregating. The four stable star structures are formed in the network. Each node is assigned a parent for data transmission in the secure hybrid structure data aggregation. N. M. S. Kumar et al. [105] Presented a hybrid model with clustered nodes forming a connected dominating set for augmenting data transmission using the machine learning technique. They also discussed dynamic energy clustering and preservation strategies to enhance the network lifetime. They further addressed the intrusion detection of DoS attacks using the hybrid model. D. B.D. and F. Al-Turjman [106] presented a secure monitoring and routing protocol using a two-fish symmetric essential approach based on the encryption and authentication model. The hybrid model was built by combining the routing and distance vector protocols. N. Rouissi and H. Gharsellaoui [107] examined a novel LEACH-based energy-efficient routing protocol on Watermarking for WSNs. They proposed a hybrid scheme based on Watermarking-LEACH for data integrity and energy efficiency. E. A. M. Anita et al. [108] Proposed a strategy to improve compromised data links between non-compromised sensor nodes. The scheme also provides high-resilience communication links against the compromised nodes in the presence of many malicious nodes.

2.8 Research Gap

Security issues and concerns linked to the design, development, and implementation of intrusion detection systems, secure localization-based blockchain technology, and artificial intelligence approaches for use in IoT-based wireless sensor networks are revealed in the literature. All blockchain-supported IoT networks are becoming more vulnerable as distributed denial-of-service (DDoS) assaults rise in the blockchain IoT ecosystem. An intricate distributed security architecture is needed to protect an IoT network using blockchain technology. They are using the appropriate analytical tools in a decentralized architecture and ensuring that a security mechanism can handle the vast amounts of data produced by IoT devices. Developing an IDS that can distinguish between benign and malicious web-based activities is challenging. However, little is known about preventing distributed denial of service (DDoS) attacks against mining pools after the relevant architecture has been implemented for blockchain-enabled IoT wireless sensor networks. In light of these challenges, we introduced a hybrid federated machine learning-based technique for discovering and pinpointing the origin of routing attacks in WSNs-IoT by examining sample datasets.

❖ **Accuracy of security localization**

It is found that the accuracy of security localization is a severe challenge in WSNs. Malicious nodes can impact the accuracy of security localization by creating fake identities and misleading the routing information. This makes gathering information about the physical or geographical locations more difficult due to the harsh weather the sensor nodes may be changed.

❖ **Physical security of sensor nodes**

Prone to the physical capturing of sensor nodes attack. Attackers perform power analysis attacks to extract sensitive information.

- **Detection of faulty nodes**

Nodes deployed in harsh environmental conditions may be failed, which further disturbs the configuration of the network. Due to this change, the network topology must cope with the different security protocols.

- **Intrusion Detection of internal and external attacks in WSNs**

They do not provide full proof of security against various attacks for multiple layers. Most of the research h finding focus on designing and planning attack-specific techniques in WSNs. Regarding intrusion detection, researchers are currently exploring using machine learning algorithms, such as Support Vector Machines, to detect internal and external attacks in WSNs. These algorithms can be applied to detect anomalies in the network traffic or abnormal behaviour among nodes. However, the dynamics of attacks change occasionally, so the topology and security performance of IDS should cope with attacking scenarios.

- **Energy and communication overheads**

Characterization of energy consumption and communication overhead is essential for security in WSNs. Energy is a significant factor in wireless sensor networks, as the sensor nodes have limited battery power. Security protocols can add to the energy consumption of the nodes, as they require additional processing power and communication. It is essential to ensure that the energy cost of security protocols does not exceed their benefits. Security protocols also add communication overhead to the network, as the nodes must exchange messages for authentication and encryption. The communication overhead can be reduced by using lightweight security protocols and optimizing the communication so that only the minimum amount of messages are exchanged.

In light of these issues, we proposed using several techniques, such as the design of intrusion detection systems (IDS), localization techniques, secure data aggregation and clustering techniques, and a secure blockchain enabled by Federated Learning (FL), in IoT-based wireless sensor networks to detect and pinpoint attacks. FL-enabled devices can learn together without ever sending data to a centralized server. This means that ML/DL can be trained in a distributed manner, wherein numerous devices and servers process data over multiple training iterations. Local learning and model transmission are the two critical components of this strategy. They allow for the same privacy protection and cost savings common in more conventional centralized machine learning systems. All ML/DL models can be improved with time using FL. At the beginning of each round, the FL server selects a subset of clients to participate in the learning process by providing them with its most recent global model.

2.9 Conclusion

The literature review reveals a need for further research into enhancing security in wireless sensor networks integrated with the Internet of Things (IoT) using different security schemes. Current research has focused mostly on cryptographic approaches to secure communication in these networks. However, further research is needed to develop novel security schemes tailored to the IoT's particular features, such as its distributed nature and limited resources. Additionally, there is a need to investigate the different security approaches to provide a comprehensive security solution. Furthermore, research into the challenges of designing secure, reliable, and robust systems for these networks is also needed. Finally, research is needed to develop secure, scalable, and efficient management systems for these networks based on various techniques for improving security and network lifetime.

IDSs are crucial to security systems and detect data theft and tampering. IDSs detect internal and external threats in WSNs. Internal attacks come from the WSN, while external attacks come from outside. WSNs utilized signature-based, anomaly-based, and behaviour-based IDSs. WSNs need secure localization. It locates sensor nodes for navigation, surveillance, and other uses. DV-hop algorithm, RSS, DE, TOA, TDOA, and angle-of-arrival are localization methods (AOA) based on range-based and range-free localization approaches. Lately, distributed and immutable blockchain-based secure localization methods have been developed. WSN security is improved through machine learning. Machine learning algorithms detect intrusion and traffic irregularities. Benchmark datasets are also used to evaluate and compare the performance of machine learning techniques. WSNs need secure clustering and data aggregation. Data aggregation and clustering group sensor nodes and forward data to the sink node. Secure clustering and data aggregation prevent hostile nodes from compromising data. Finally, WSN security requires trust management and routing for distributed and hierarchical topologies. Trust management methods detect and authenticate legitimate nodes, whereas routing methods securely and effectively route data. Trust and routing methods play crucial roles in ensuring the secure and efficient operation of the network.

Chapter 3

3 DESIGN OF SECURE INTRUSION DETECTION SYSTEMS

3.1 Introduction

wireless sensor networks (WSNs) are Affordable and low-power sensor nodes [109]. Sensor nodes with multi-hop routing and self-organizing intelligent sensor networks fall into this category [110].WSNs are autonomous spatially disseminated devices using sensors for physical and environmental conditions. WSNs are self-configured and connected by radio signals with a low operating battery and low cost distributed hierarchically and randomly [105][29]. WSNs are recent technology and have gained significant attention for research scenarios [29]. They comprise low-power and cost sensors randomly distributed over the target localization. The sensors are distributed to act on specific tasks [111], [112]. The sensors have sensing and signal processing capabilities and activate wireless communication in WSNs [113]. Wireless sensor nodes have limited computational processing, battery, and communication capacity. In WSNs, a gat way provides wireless connectivity using wired and distributed nodes. WSNs are the main components of the Internet of Things (IoT), with small, low-power sensors for data collecting and monitoring from the environment.

3.1.1 IDS-Based Attack Detection

Cyber-attacks on international businesses are on the rise, and in response, industry and academic research are rapidly developing intrusion detection systems (IDS) [114]. The intrusion detection method is an intrusion detection technology, a network security protection scheme for collecting and analyzing network data to detect abnormal behaviour [115]. Two types of intrusions can be identified in a WSN system: anomalies and misuse. As a mathematical model, anomaly detection follows a regular behaviour profile [115] and uses estimated feature values different from the average reference values [116] for network models. A cut-off point establishes it. Methods for detecting anomalies can be found in data mining, cluster analysis, and machine learning. Alternatively, the Misuse detection process is educated by information about known instances of harmful activity and the unique infiltration patterns associated with those instances. Therefore, the infiltration is identified by comparing designs to the data store. The causes for these cybercrimes are DoS and web attacks. The attacks in WSNs can be active or passive based on their activity in various network layers. Intrusion refers to active and passive unauthorized eavesdropping and information gathering in a network via damaging packets by dropping, forwarding, and hole assaults [117]. Protected WSNs rely on two layers of defence: intrusion detection and prevention technologies. The primary functions of IDS [117] are intrusion detection and providing facts about the intruder to the system administrator. Intruder, time, activity, location, category, and network layer are all tracked by these. It's essential for network security [118]. Cybercrimes like this harm businesses because they introduce harmful attacks into the networks. To fight back the attacks in WSNs, detection of intrusion, which is a vital part of cyber security, identifies malicious network activity. Intrusion detection is the examination and identification of security breaches in an information system.

The intrusion detection method is an intrusion detection technology, a network security protection scheme for collecting and analyzing network data to detect abnormal behaviour [115]. Attack attempts, actions, and consequences are detected through monitoring network and computer system operations. And then immediately issue an alarm or take action to protect system resources. Network intrusion defence relies on intrusion detection systems. Two types of intrusions can be identified in a WSN system: anomalies and misuse. As a mathematical model, anomaly detection follows a regular behaviour profile and uses estimated feature values different from the average reference values for network models [115], [116]. A cut-off point establishes it. Anomaly detection techniques are used in machine learning, data mining, and clustering. At the same time, The Misuse detection mechanism is informed by data on previously observed malicious activities and the specific infiltration patterns associated with them. Therefore, the infiltration is identified by comparing designs to the data store.

The causes for these cybercrimes are DoS and web attacks. The attacks in WSNs can be active or passive based on their activity in various network layers. Intrusion refers to active and passive unauthorized eavesdropping and information gathering in a network via damaging packets by dropping, forwarding, and hole assaults [117]. Protected WSNs rely on two layers of defence: intrusion detection and prevention technologies. The primary functions of IDS [117] are intrusion detection and providing facts about the intruder to the system administrator. Intruder, time, activity, location, category, and network layer are all tracked by these. Cybercrimes like this are extremely harmful to businesses because they introduce harmful attacks into the networks. To fight back the attacks in WSNs, detection of intrusion, which is a vital part of cyber security, identifies malicious network activity. Intrusion detection is the examination and identification of security breaches in an information system.

3.1.2 Machine Learning Techniques

The power of ML systems rests in their capacity to offer generic answers via a learnable architecture that may continuously enhance efficiency [119]. Having such broad applicability across disciplines, it is essential in areas as diverse as engineering, medicine, and computer science. New methods from the field of ML have been put to use in WSNs to address a wide range of problems. Using ML boosts WSN performance and reduces the need for manual maintenance and re-programming. It is difficult, if not impossible, to access the massive amounts of data produced by sensors and extract the relevant information without the aid of ML. It's also used to combine M2M communications, cyber-physical systems, and the Internet of Things (M2M). Here are a few ways that ML can be used in WSNs:

- The ideal number of sensor nodes to use to cover a specified area is determined using ML algorithms.
- With the help of energy harvesting, WSNs used in harsh environments can function without external power sources.
- Improved performance and security of WSNs in predicting the amount of energy to be gathered within a given time slot is a direct result of the application of ML algorithms.
- The locations of sensor nodes can shift for various reasons, both internal and external. With the help of ML algorithms, precise localization is swift and painless.
- ML was employed to distinguish between good and bad sensor nodes to increase the network's performance.
- Improved network longevity is mainly attributable to routing data. A dynamic routing technique is necessary to improve the system performance due to the unpredictable nature of sensor network behaviour.
- Computing the positions and locations of wireless sensor nodes in two and three dimensions enables the localization of malicious sensor nodes. The sensor nodes used in many WSN

systems are deployed in the field without prior knowledge of their locations, and often, insufficient infrastructure is in place to track them down afterwards.

- By extracting routing attributes that will aid in routing discovery in WSNs, ML approaches are useful for detecting and classifying DDoS attacks across multiple layers of sensor networks.

3.1.3 Clustering and Aggregation

Sensor nodes are clustered into groups with distributed energy loads to increase the network's lifetime [29]. They communicate with limited interference using a direct sequence spread spectrum. The LEACH protocol is used for data collection and clustering nodes in hierarchically distributed WSNs. Figure 3. 1 shows a network model comprising the base station, cluster head, sensor nodes, and malicious nodes. This protocol creates a clustered topology for reducing the level of energy consumption. The selection of cluster head is rotational from the distributed node. The clustered structure is formed by LEACH, classified into circles utilizing a TDMA schedule. Each initializes the processes and creates a data frame for transferring aggregated data to the sink node. The LEACH protocol utilizes fuzzy rules for identification in different classes of attacks, including Sybil attacks, wormhole attacks, and hello flood attacks. The network model uses Swarm intelligence and optimized algorithms for optimal path selection and solving routing problems [120]. The whole data transfer to the sink node utilizes the aggregator node [111]. The aggregator node, referred to as the cluster node, solves the problem of large-size data in energy-constrained WSNs. The cluster node reduces the energy consumption by the information aggregation technique from the sensor nodes to the sink node. Machine learning strategies are suggested for selecting the cluster head for enhancing energy performance. ML techniques are used to detect and remove malicious nodes at the cluster head (CH). The CH utilizes the ML algorithms for effective energy consumption and enhances the network lifetime. The cluster head acts as a local base station sensor for data aggregation and forwarding data to the sink node.

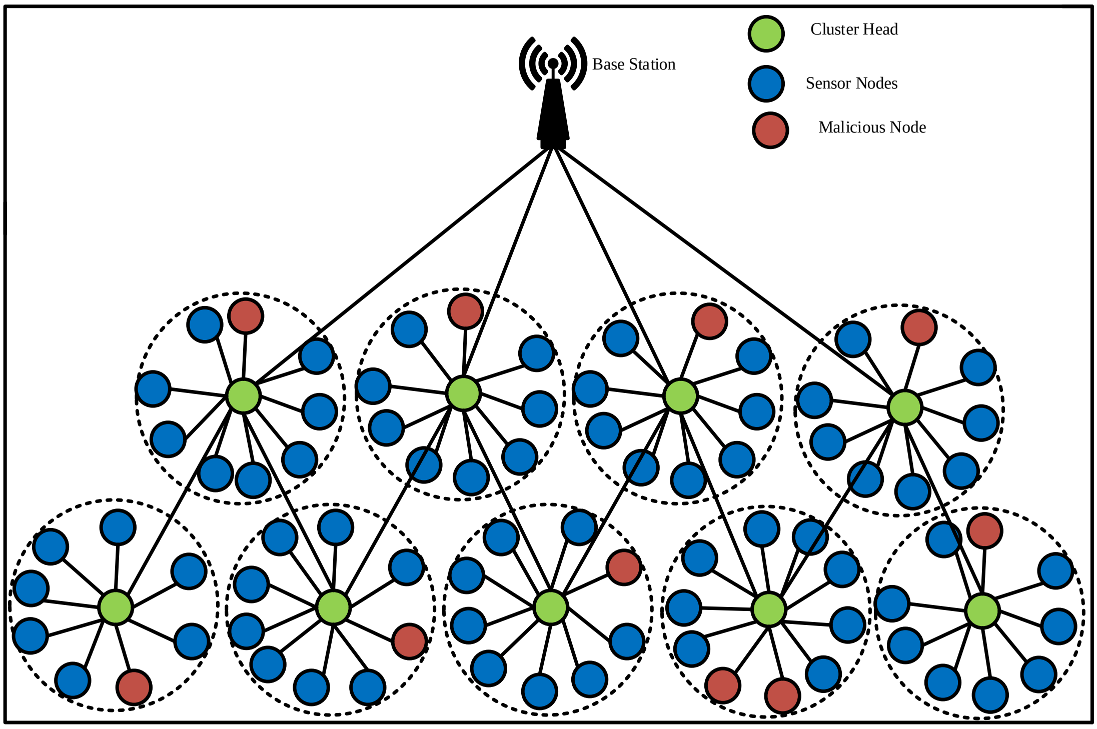


Figure 3. 1. Wireless sensor networks model with clustering Nodes using a routing protocol.

Data aggregation gathers and combines information from various sensor nodes [121]. In WSNs, data aggregation can impact many metrics, including energy consumption, storage requirements, network load, and processing speed. Data aggregation is crucial to lessen the transmission and communication burden in WSNs. Combining data from multiple sensors can improve the network's reliability and lifespan. Only a handful of possible data aggregation methods are cluster-based, tree-based, in-network, or centralized. The security and efficiency of WSNs can be improved in several ways, including clustering and aggregating data. Here are some examples of ML algorithms used in WSNs to collect and organize data, along with the advantages they provide:

- The machine learning methods used to select CHs for data aggregation in the network significantly balance the energy consumption of the sensor nodes.
- Data dimensionality can be reduced at the sensor node level with the help of ML algorithms, which lowers the network's communication overhead. The reduction can be made during the data collection and aggregation stages to speed up data transmission.
- In the context of data aggregation, ML may adjust to its surroundings without requiring any additional configuration or reprogramming.

This chapter is divided into distinct parts for easier readability and comprehension. In the first section, we provide a high-level overview of wireless sensor networks, sensor clustering, our inspiration for doing this study, and the results so far. Figure 3. 2 shows the sections of organization and framework of the proposed system.

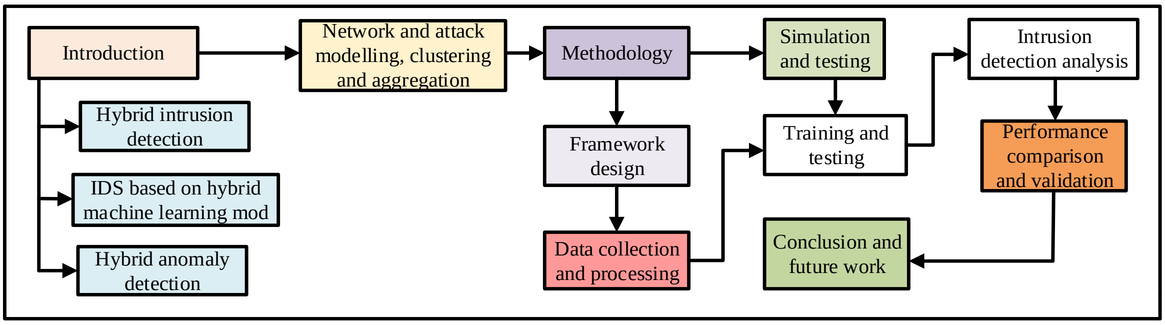

Introduction
Hybrid intrusion detection
IDS based on hybrid machine learning mod
Hybrid anomaly detection
Network and attack modelling, clustering and aggregation
Methodology
Framework design
Data collection and processing
Simulation and testing
Training and testing
Conclusion and future work
Intrusion detection analysis
Performance comparison and validation

Figure 3. 2. Organization and framework of the proposed work.

3.2 Problem Statement

Hierarchical WSNs consist of multiple layers or levels of sensor nodes, enabling efficient data aggregation and transmission in remote and unmonitored sites. Due to this, several types of attacks can target WSNs, including black holes, Sybil attacks, sinkholes, wormholes, forwarding attacks, and grey holes. These attacks exploit vulnerabilities in WSNs and can compromise the security and privacy of the networked infrastructure.

To address these security threats, we proposed Advanced Intrusion Detection Systems based on hybrid machine learning (AIDS-HML) to enhance security in WSNs. These systems employ hybrid machine learning classifiers to identify and classify attacks in wireless sensor networks. We also simulate the routing attacks and evaluate the system performance. The proposed model is compared with baseline models using benchmark datasets, and evaluation metrics such as energy, time, precision, recall, F1-score, and accuracy are used to assess the performance of the models.

3.3 Network Models and Clustering Techniques

The base station, cluster head (CH), and sensor nodes comprise the hierarchically distributed wireless sensor networks (SN) network paradigm. In this setup, the sensor nodes use a wireless connection to communicate with the cluster node and the sink node [5]. When designing and planning the network model, the following assumptions are incorporated as shown below:

- Every mobile sensor node can roam freely within the network area [122].
- The sensor nodes are deployed randomly.
- All of the sensor nodes are the same in every way.
- Any location within the network's range is possible for a sink to be installed.

As a result, the position of unknown nodes in the network can be calculated with the aid of beacon and sink nodes, both aware of their position and location. When it comes to routing assaults, the WSN's nodes are unprotected. In most cases, this kind of attack shortens the lifespan of the sensor nodes and causes them to run out of juice. Tunnels created by routing assaults distort the route path and use routing resources. To protect against denial-of-service and routing attacks, the proposed network model incorporates node-level security measures. The proposed system employs state-of-the-art intrusion detection systems based on hybrid machine-learning algorithms to detect and localize cyberattacks.

The CH is powerful in processing, computation, and battery life. Over a predetermined time, the sensor nodes compare their findings and report back to the cluster node with their collective verdict. In this network model, the sensor nodes are supposed to be similar in all respects. Positions of sensor nodes are organized in hierarchies and clusters. The data is transferred from the source nodes to the sink node via the cluster nodes [123]. Figure 3. 3 shows that the suggested network model employs five types of nodes: the sensor node, malicious nodes, central nodes, cluster nodes, and sink nodes.

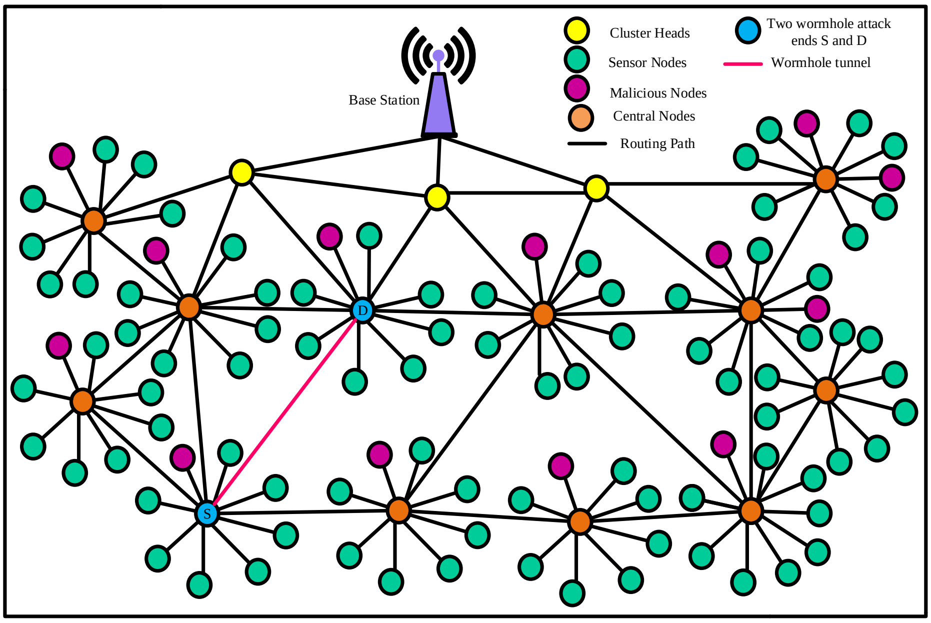


Figure 3. 3. Hierarchical topology and configuration model for secure wireless sensor networks [22].

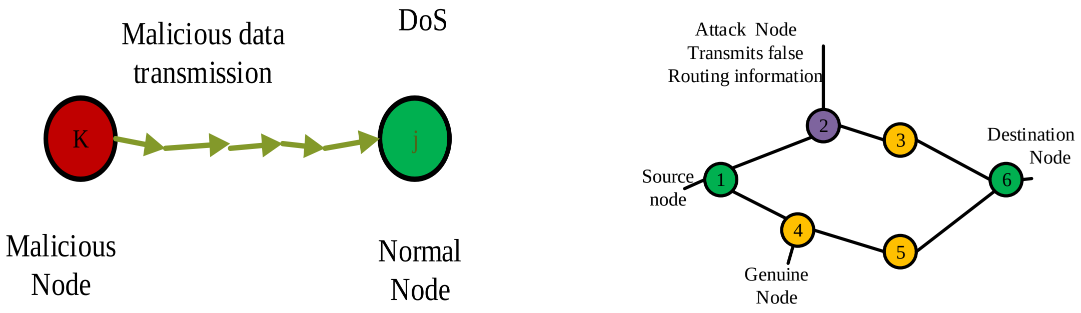
The function of the cluster head (CH) collecting network traffic and relays it to the base station. It employs straightforward reasoning based on a binary classification approach to attacks employing decision tees. CH is selected in a trustworthy and once-per-round-trip process, and the network is safeguarded. In this distributed network concept, all sensor nodes are identical and spread out randomly, while the base station is a stationary power source. The cluster head connects all sensors and functions as a root node to stop malicious communication by applying binary classification with a threshold value.

Each cluster group relies on the central node and CH to relay information to the BS node and aggregate it for analysis [124]. The CH also avoids depletion energy using the isolation table for attack detection. Two primary and secondary cluster heads for intrusion detection of attacks. The attack models provide a graphical representation of the network topology, along with details about key identities and routing information that can be used to identify and exploit security flaws in the system. Figure 3. 4. Jamming (a) and Sinkhole (b) attacks at the physical and network layers [22].

It shows that it is presumed that they have limited means and intellect to interrupt network traffic. Several variants of WSN attacks are simulated to test the proposed IDS's effectiveness. The attack model measures how well and securely the system functions. The network's threads are defined by the application layer, the Media Access Control layer, the Physical layer, the Transport layer, and the Network layer [125]. Jamming security threats sends harmful data, disrupting short-range connections. The transmission of jamming signals causes legitimate user and service blocking. Following is a mathematical model of an attack as in equation (3.1):

$$I_a = \sum_{i=1}^{n} e_i + \sum_{i=1}^{n} m_i \tag{3.1}$$

Where I_a is the transmitted information depending on the IDS that can be correct or incorrect, e_i is the expected information, and m is the malicious content information [110]. The data is detected as malicious nodes have considered the network's transmitted data and energy auditing. The channel priority is the major factor in the medium access control layer. The malicious nodes modify and change the back-off time using the manipulation approach. The attackers advertising false information in the network affects the layer routing information like the minimum hope count.



(a) Jamming attacks (b) Sink hole attacks

Figure 3. 4. Jamming (a) and Sinkhole (b) attacks at the physical and network layers [22].

3.4 Methodology

These are general stages to model and simulate a system for producing and extracting datasets. Since getting real datasets in WSNs is difficult, we use standard datasets to evaluate the effectiveness of the new advanced intrusion detection system based on hybrid machine learning models. The usefulness and efficiency of the proposed enhanced intrusion detection method based on hybrid machine learning on various classes of assaults are demonstrated by utilizing publicly available datasets [126]. The KDDCup 99, NSL-KDD, UNSW_NB15, CSE-CIC-IDS20182, and CICIDS2017 datasets are extensively used for academic study and research for attack detection evaluation as benchmark datasets. The system's effectiveness is tested using the KDD Cup 99 datasets [127], [128] and the intrusion detection techniques. These datasets aim to measure the IDS using a predictive decision model. The 1999 DARPA dataset is also used in this work. The dataset is evaluated using offline and real-time evaluation modes. The data is processed in various modes so that the usual behaviour of the network may be determined.

3.4.1 Benchmark Datasets

Some issues with the KDD'99 dataset are discussed, and a dataset called NSL-KDD is proposed as a solution [129]. Although it is a new and standardized genre of the KDD data set, it still suffers from some of the problems studied by McHugh. It may not be a perfect illustration of existing real networks. Still, it is used and applied effectively as a standard data set to help researchers compare the various network-based IDSs. In addition, the NSL-KDD training and testing sets have a manageable quantity of records. The cost of doing trials on the entire dataset, as opposed to a sample, is reduced by using this method. By combining the KDD'99 Data Set with the NSL Data Set, we get several advantages over the original KDD data set:

- It avoids training classifiers on duplicate or redundant records by omitting them from the train set.
- The proposed test sets don't reuse any records; thus, the approaches with higher detection rates on common data won't unfairly boost the learners' performance.
- The proportion of records in the original KDD data set is inversely proportional to the number of records chosen from each group of challenging levels.
- Since there aren't many records in the train or test sets, we can afford to perform the experiments on the whole set without randomly picking a subset.

The NSL-KDD dataset is also used as a benchmark to test the detection performance of the proposed system using semi-supervised machine learning [121] models for a class of attacks with 42 attributes and class labels. Forty-one attributes are classified into content, host, traffic, and basic features. The dataset has a total record of 148,515 samples sectioned into 80% of training and 20% of testing samples, as shown in Table 3. 1, with four different classes of attacks. The vector features are extracted for training by splitting the dataset into clusters as normal and abnormal. After training, the vector features are received for classification as normal and abnormal clusters.

Table 3. 1. Frequency distribution of various attack classes in the NSL-KDD with training and testing samples for testing performance.

samples	Normal	DoS	Probes	U2R	R2L	Total
Training	67,343	45,927	11,656	52	995	125,974
Testing	9,711	7,458	2,421	200	2,654	22,544
Total number	77,054	53,385	14,077	252	3,649	148,518

The dataset consists of 23 classes of attack types and is clustered into four classes of attacks, including denial of service (DoS), remote to local (R2L), user to root (U2R), and probe category. The DoS attack makes the network service busy and the authorized user inaccessible from the network [117]. The U2R attack applies vulnerabilities to the host system by sniffing the passwords of the legitimate user. The R2L injects vulnerabilities remotely into the system of the network host. The probe attack scans the network for information collecting and gathering, violating the security rule. The probe and DoS attacks have multiple links, whereas the others have single links [130]. Table 3. 2 shows the description of the four classes of attacks in the NSL-KDD benchmark dataset.

Table 3. 2. Provides a detailed technical description of the four types of attacks included in the dataset.

Attack	Attack description in the dataset
DoS	The attacker makes the network busy and denies the legitimate user access.
R2L	The intruder tries to gain access to the network for a specific version of the FTP.
U2R	The attacker accesses the system's root and makes unauthorized attempts to the network.
Probe	It endeavours to assemble the data behind evading the security of the system.

The UNSW_NB15 dataset is used as a benchmark for evaluating the effectiveness of the proposed system. This dataset has hybrid synthesized attack activities and normal network traffic data [131]. The IXIA traffic generator is arranged with three virtual servers for generating the UNSW_NB15 dataset containing normal and malicious activities in the network traffic. The servers are established using public and private network traffic having IP addresses with routers. The routers are configured with a firewall that filters the traffic as normal and malicious activities. The tcpdump tool is installed on routers for capturing from the IXIA tool dispersed among the network nodes utilized as attack traffic generators with normal network traffic. The frequency distribution of the class of DoS attacks is shown in Table 3. 3 with training and testing samples.

Table 3. 3. Frequency distribution of attacks in the dataset.

	Training	Testing		Training	Testing
Attack category	weight	weight	Attack category	weight	weight
Analysis	677	2000	Generic	18871	40000
Backdoor	583	1746	Normal	37000	56000
DoS	4089	12264	Reconnaissance	3496	10491
Exploits	11132	33393	Shellcode	378	1133
Fuzzers	6062	18184	Worms	44	130

The method's effectiveness for identifying flooding assaults in WSNs is measured against the CIC-IDSS2017 dataset. Table 3. 4 details some of the key elements of the training and testing dataset found online at the Canadian Institute for Cyber Security Research LAB. Data about network traffic, both benign and malicious [132], is included in the dataset. It was manufactured to serve as a plausible in-the-background activity while gathering data. Twenty-five individuals utilizing a variety of protocols were used to compile the dataset.

Table 3. 4 shows that the dataset used to develop the predictive machine learning models contains 485881 occurrences and 31 characteristics divided into training 80% and 20% testing subsets. There are five distinct varieties of DoS attacks [133], including the widely-known Slowhttptest, Slowloris, Hulk, Heartbleed, and GoldenEye. DoS attack samples are used for training and testing.

Table 3. 4. Structure of the dataset with classifications of assaults.

Attacks	Slowhttptest	Normal	slowloris	GoldenEye	Hulk	Heartbleed
Training set	1276.8	252382.6	1504.8	6307.2	127302.4	8
Testing set	319.2	63076.4	376.2	1576.8	31825.6	2
Overall	1596	315382	1881	7884	159128	10

Normalization, missing value imputation, and aggregation are all part of the data processing required to rearrange the data before the training and testing phases begin. We fill in the blanks by averaging the current values [134]. It is possible to convert the data into binary values as 0 and 1 by using the minimum and maximum values.

3.5 Proposed AHIDS Framework

The first phase of the proposed system for an advanced hybrid intrusion detection system (HIDS) in WSNs is the deployment of sensors, as shown in Figure 3. 5. This involves strategically placing the sensors in an optimal position to monitor the entire network and maximize coverage. The sensors should be placed efficiently, minimizing power consumption and maximizing coverage. Additionally, the sensors should be robust and reliable to monitor the network for any malicious activity properly. After sensors are deployed, a simulation tests system performance and sensor placement. Optimization improves system performance and accuracy after simulation. This includes tweaking detection techniques, sensor placement, and machine-learning model parameters. The optimization process helps choose the best WSN routing algorithms. This reduces data transfer size and increases system efficiency. Machine learning algorithms detect and classify risks to protect data. Machine learning models detect and classify assaults using labelled data. After detecting threats, the system can respond. Alerts and stopping harmful activity are examples.

The figure depicts the framework used in the proposed system to categorize and classify network data flow as either normal operations or harmful attacks. As a result of its ease of use and superior performance in hierarchical clustering wireless sensor networks, the suggested technique is likely to find widespread use. Conditional control statements are a key component of the hybrid machine learning approaches for decision-making events' outcomes. The new aspect of this decision-making method is the use of a collaborative process for data analysis, which in turn aids in the automatic construction of predictive models. Decision nodes are used for prediction, while leaf nodes are used for the final classification, as seen in Figure 3. 5 using hybrid machine learning models.

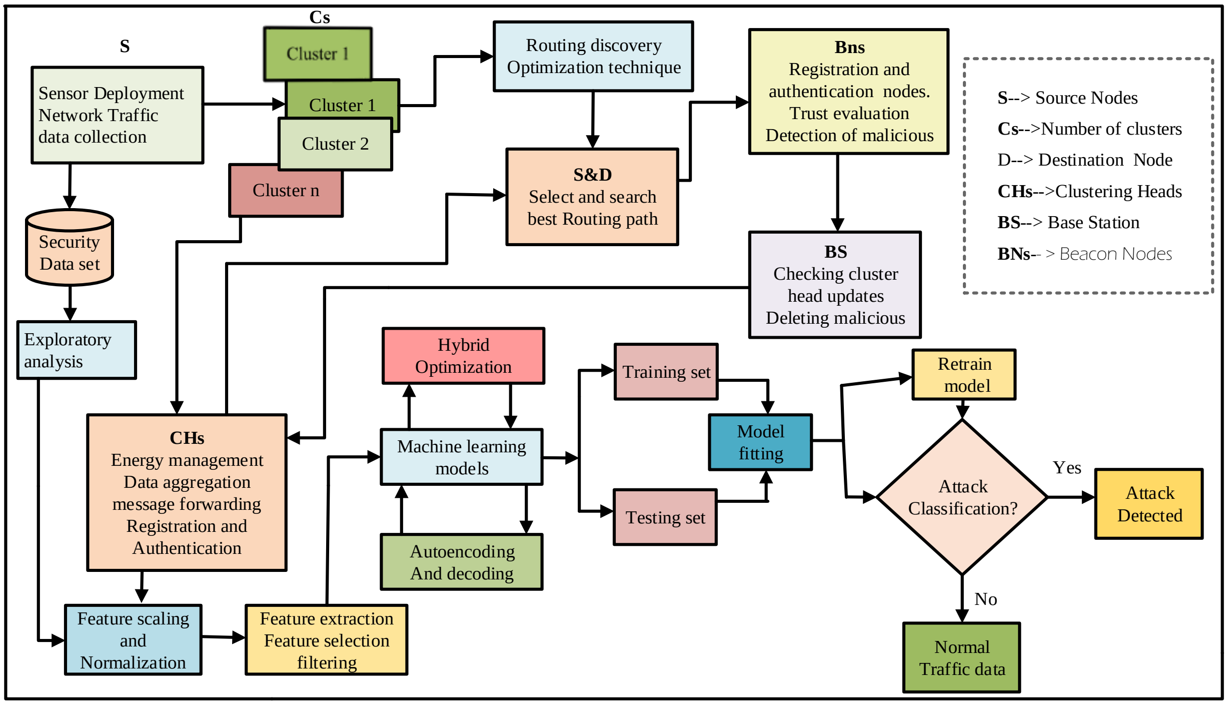


Figure 3. 5. Framework for advanced hybrid intrusion detection system (AHIDS) Block diagram using attack detection and classification Model.

Splitting training and testing benchmarks is governed by rules and reasoning generated by the hybrid machine learning models. Target categorization is performed using the statistical metric. Finally, the proposed system employs hybrid machine learning approaches [135] to detect and localize attacks utilizing data from both the attack and non-attack phases. Using the modified dataset, the suggested AIDS-HML learns to identify potential attacks. Assuming that all features of every sample belong to the designated class label, AIDS-HML is an efficient classification method (conditional independence assumption). Enhanced Advanced Hybrid Machine Learning (AIDS-HML) is a more advanced and hybridized version of machine learning.

3.5.1 Sensor Deployment and Routing Techniques

Sensor nodes are deployed based on various network models for attack detection and classification attacks in WSNs. However, WSNs face significant difficulties in routing because of their restricted power supply, poor transmission bandwidth, less memory capacity, and processor capacity [121]. Due to limitations like short battery life, small memory, and low processing power, an adversary can quickly target individual nodes of WSNs when deployed in a dangerous area [112]. It is crucial to identify malicious attacks to prevent being tricked by the adversary's fabricated data supplied by compromised nodes. Here, we distinguish between internal and external attacks on WSNs. The goal of the external attack is to reduce the effectiveness of the WSN and is carried out by parties outside of the network. Therefore, we shall elaborate on the proposal that protects against routing attacks with data while maintaining its integrity. The proposed scheme utilized HML methods for WSNs to create safe protocols for extracting features and locating new routes in moderately complex hybrid and tree network topologies. The following are a few of the many advantages that machine-based routing brings to WSNs:

- Without requiring re-programming, ML algorithms can adapt to new environments and select new CHs for routing in WSNs.
- HMLs can be used for various purposes in WSNs, including optimal routing, reducing communication overhead, and delay awareness.

In this study, we employ the GA-ANN method for detecting wormhole assaults and determining energy-efficient and robust routing for WSNs. WSNs utilize GA-ANN to train their protocols based on a wide range of inputs, including residual energy, node distances, routing discovery, path selection, feature extraction, cluster heads (CH), border nodes, and the sink or base station. An enormous training set is produced, and even ANN is provided with effective threshold values for picking a set of trustworthy CH via backpropagation. Data loss in WSNs can be prevented, and energy consumption among sensor nodes is balanced with this technique.

3.5.2 Data Pre-Processing

A raw dataset may not have undergone any pre-processing [136]. A raw dataset is incomplete, noisy, and possibly presented unfavourably. Therefore, it is not viable to construct machine learning models from scratch with a raw dataset. Figure 3. 6 shows data pre-processing techniques for the training and testing machine learning models. Pre-processing the raw data by removing irrelevant information and standardising the format can increase the effectiveness of a machine learning model. It is for this reason that this stage of a machine learning model is the most crucial. With the Data collection module, you may get traffic statistics from either a private network or the reference data set [137]. The data are sent to the pre-processing module, cleaned, and prepared for use. This information is also provided to the cluster and the trust-based safer routing module for better data transfer. The agent that does data pre-processing makes great use of the tools. Before the pre-processing phase, data is cleaned, integrated, and modified. Cleaning and normalizing the data is the first stage in the data pre-processing phase, which aims to boost the data quality used in training and testing to develop machine learning models for prediction. Pre-processing, which includes feature extraction, feature selection, and dimensional reduction, is crucial for vectorizing [138]. Duplicate values can be removed, the mission data can be replaced, and unneeded sample structures can be eliminated. Normalization using minimum and maximum scaling values, as in equation (3.2), must be done after the dataset has been cleaned.

$$Z_{\text{norm}} = \frac{Z - \min(i)}{\max(Z) - \min(Z)} \tag{3.2}$$

Where min(z) is the minimum value and max(z) is the maximum value of the attribute Z, respectively. Z_{norm}, is a normalized feature value, and Z is an original feature value [139].

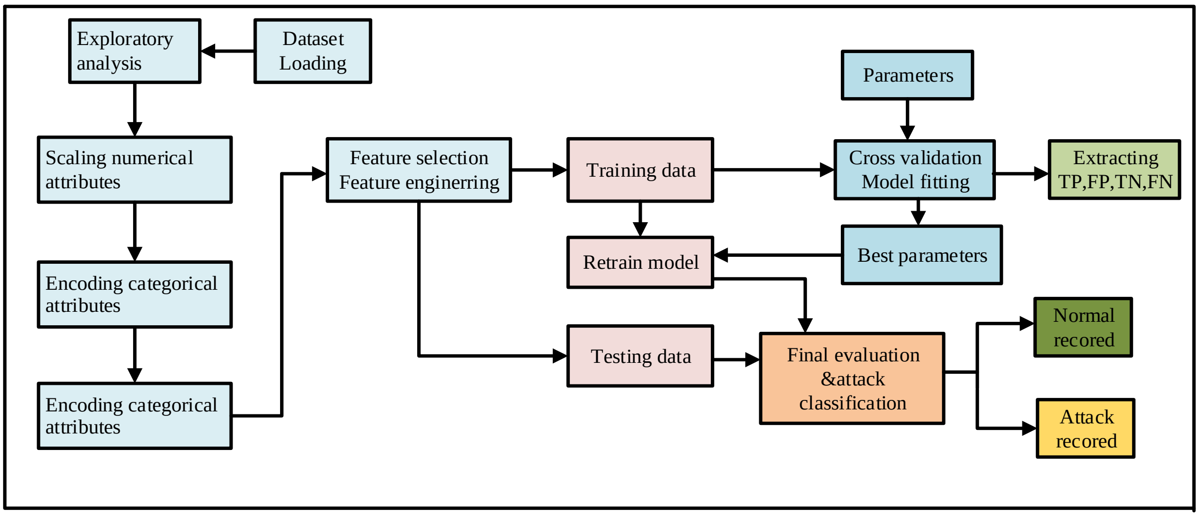

Exploratory analysis
Dataset Loading
Parameters
Scaling numerical attributes
Feature selection Feature enginerring
Training data
Cross validation Model fitting
Extracting TP,FP,TN,FN
Retrain model
Best parameters
Encoding categorical attributes
Normal recored
Testing data
Final evaluation &attack classification
Encoding categorical attributes
Attack recored

Figure 3. 6. Data pre-processing, training, and testing for model evaluation framework using benchmark datasets.

K-means cluster sampling improves the machine learning model's classification and detection performance by creating a tiny K-number of clusters from the original dataset to decrease the training complexity [41]. By eliminating unnecessary information, the K-means sampling method improves productivity, computational power, and resource use by producing highly representative small groups. Synthetic minority oversampling (SMOTE) generates superior samples to address inequalities between demographic groups. When the data has been preprocessed, feature engineering creates sensitive, high-quality features, minimizes dimensionality, and eliminates redundant features by computing the correlation features between them.

Filtering: One way to clean up data is through filtering. This filter roughly estimates a desired signal pattern from a distorted signal pattern. The major goal of this filtering method is to minimize the mean square error between the estimated and intended signal patterns.

Feature Selection: Feature selection approaches choose the most relevant and useful features for model learning. This method enhances prediction accuracy while decreasing the prediction model's overfitting, training time, and complexity. There are a few different ways to pick features, but the most common ones are Filtering, Wrapping, and embedding. Figure 3. 7 shows selected features using the NSL-KDD benchmark dataset. Choosing the right features to analyze is crucial to any data mining project [140]. Finding the most effective feature selection approach and incorporating it into the appropriate procedures is crucial for the application's success.

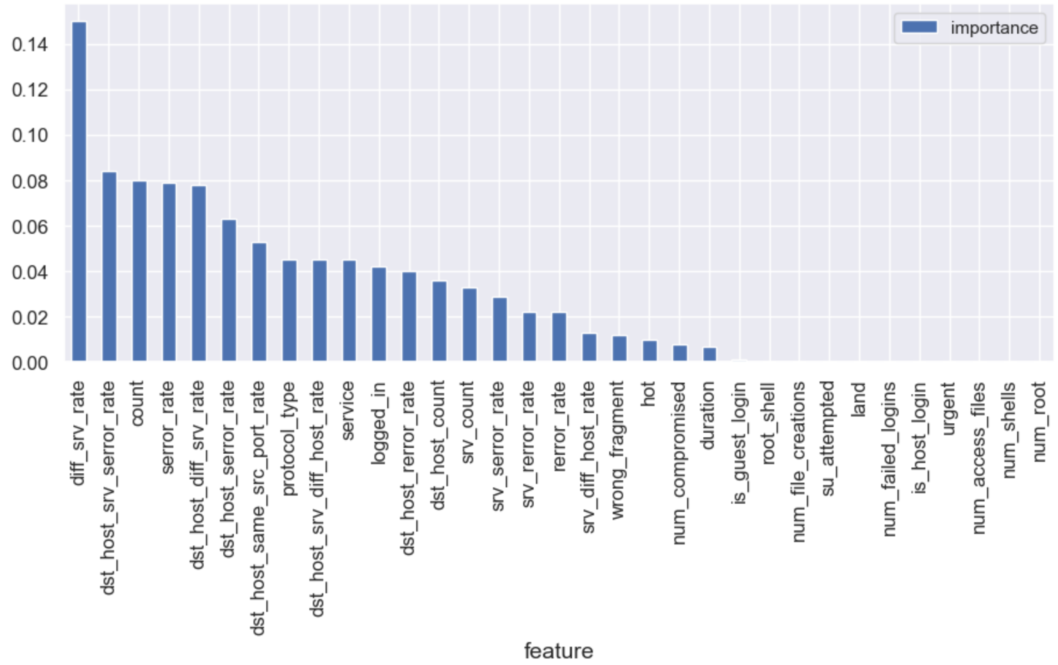


Figure 3. 7. Selection of important feature technique using NSL-KDD benchmark dataset.

Reducing the total number of attributes in the dataset and making new associations between them make the procedure less labour-intensive. Yet feature selection does not have a single, universal approach. The dataset's current condition should be considered while deciding which approach to use. Finding the best feature for discriminating between classes is the primary challenge in feature selection. Various data sets may call for distinct feature selection strategies. The feature selection method employs a plethora of different methods. Spearman's rank correlation coefficient formula is used for a recursive feature selection process, which dynamically selects features as shown in equation (3.3)

$$\rho=\frac{\sum_i (x_i-\bar{x})(y_i-\bar{y})}{\sum_i (x_i-\bar{x})^2 (y_i-\bar{y})^2} \tag{3.3}$$

Where ρ is the correlation coefficient, x_i and, y_i are the feature variables, and $\bar{x}$ and $\bar{y}$ are the mean values of x and y.

Feature engineering: Each pattern can be isolated with the help of a single clue provided by the feature engineering phase. When a raw dataset has a large feature set considered redundant, the feature extraction approach creates a derived set of non-redundant and informative features from the original feature set.

Windowing: Each pattern can be isolated with the help of a single clue provided by the feature engineering phase. When a raw dataset has a large feature set considered redundant, the feature extraction approach creates a derived set of non-redundant and informative features from the original feature set.

Filter Method: The filter technique employs feature ranking methods for feature selection. Ranking features indicate how crucial they are when constructing a model. The results of numerous statistical tests are used to rank the features. Each feature's connection with the result variable is calculated in these statistical analyses. Pearson's correlation, Linear discriminant analysis, Analysis of variance, Chi-square, Variance threshold, Information gain, and so on are only some of the many statistical tests available.

Transformation and Normalization Operations: The NSL- KDD dataset has quantified the nominal values. Attributes such as protocol type, service, and flag numbers in the dataset have had their numeric values transformed. This method relies on using purely numerical values in the dataset and the subsequent categorization processes: the names and their corresponding numbers in the conversion system. TCP, UDP, and ICMP protocols are translated to the numbers 1, 2, and 3, while the service and flag properties are translated to the corresponding numbers 1 and 2, respectively. Dataset normalization is another very significant pre-processing technique, especially in classification. The dataset's properties are normalized so that they all have consistent values. Normalization expedites operations on the dataset and increases the frequency with which they succeed. Minimum and maximum normalization is used to standardize the dataset used in the research. The method benefits from the extremes of the data set. Information is standardized to these standards. The goal is to split the remaining data between 0 and 1, with the lowest number set to 0 and the largest set to 1.

3.6 Machine Learning Techniques

Learning is iteratively using mathematical models on data to help a machine find latent patterns that can then be used for prediction [141]. Machine learning techniques are algorithms and methods that enable computers to learn from data and make predictions or take actions without being explicitly programmed. These techniques allow machines to automatically discover patterns, relationships, and insights from large datasets, leading to intelligent decision-making and problem-solving capabilities. Many different machine-learning techniques are used to construct models for solving common problems.

3.6.1 Random Forest

When several decision trees are trained using many different data sets, the resulting algorithm is called a random forest algorithm (RF). Breiman created this multi-technique classifier back in 2001 as an algorithm. Sub-training clusters are generated in the random forest algorithm. When forming a training cluster, preloading is used. In order to grow the trees, we employ a mechanism in which the attributes are randomly picked. The algorithm works by picking a random value from each node and utilizing that as the basis for a branch. Randomly selected factors produce the derived trees. The collected datasets are utilized as input into the Classification and Regression Trees (CART) algorithm for tree building. Each created tree-labels the training sample and the classes assigned to which the sample is then compiled. To be processed instances are often included in the most common classification to which they belong. The RF method does not include pruning, while the CART algorithm does. An important reason why the RF algorithm outperforms the other decision tree approaches is that it doesn't rely on pruning. The RF algorithm is quick, flexible, and more effective than alternative decision tree approaches despite its usage of numerous tree topologies. The CART algorithm uses the GINI index value to decide which branch to create from each node. Tree development parameters include the number of trees and the number of variables per node. The RF algorithm's basic operation is depicted in Figure 3. 8 for attack detection and classification using training and testing the benchmark datasets.

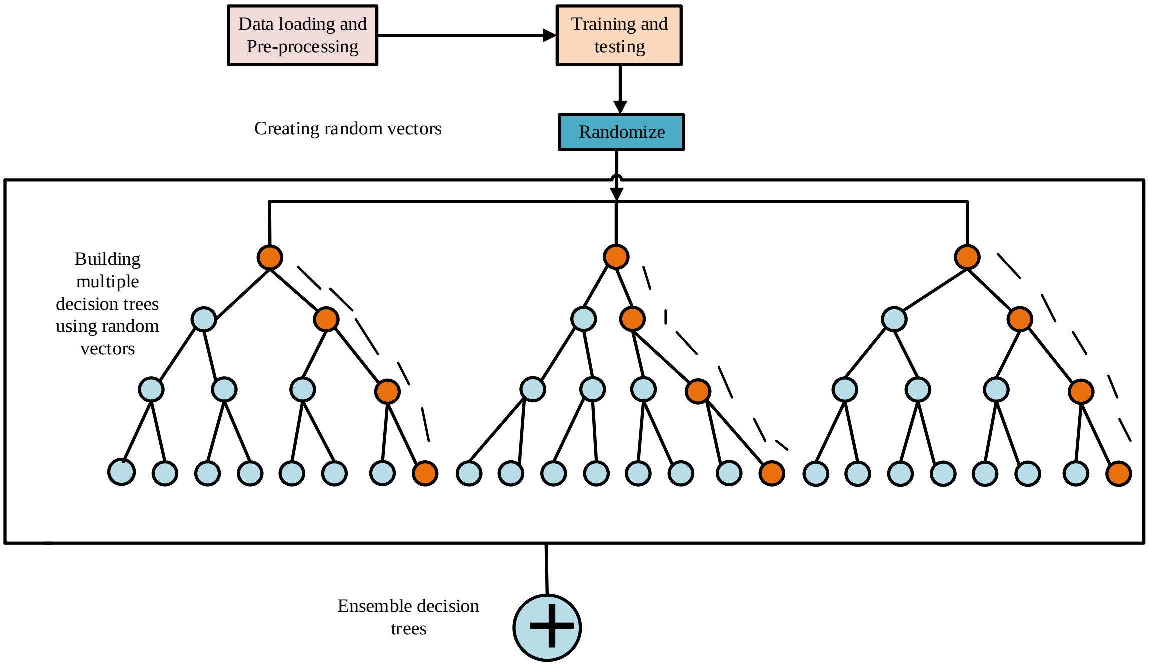

Data loading and Pre-processing
Training and testing
Creating random vectors
Randomize
Building multiple decision trees using random vectors
Ensemble decision trees

Figure 3. 8. Block diagram of random forest operation for training and testing benchmark dataset [38].

Random forest (RF) is a machine learning classifier consisting of a group of trees for building a classification predicting model [119]. The RF technique is effective for two steps, creating a random forest classifier and predicting results [142]. RF technique is effective for a large and heterogeneous dataset to predict missing values accurately. Random forest randomly selects training samples and isolates variables for each tree node to produce more decision trees. Random forest is an appropriate classifier for hyperspectral data for solving coverage and medium access control (MAC) protocols in WSNs. It is robustly applied for highly correlated data and high-dimensional data. When dealing with high-dimensional data, Random Forest handles it by evaluating the significance of features to solve overfitting and stability issues by lowering variance [136]. Because of this, the random forest method is useful for identifying and categorizing malicious assaults in wireless sensor networks utilizing a diverse benchmark dataset. Also, the random forest can deal with missing data and is resistant to anomalies without too much impact on the rest of the data. Due to the extensive tree-building involved, the plan also necessitates greater computing power and additional resources. To forecast the detection accuracy of the model, Random Forest builds a forest out of many decision trees [143]. It can recognize and classify threats using a common attack dataset.

3.6.2 Decision Trees

Decision trees (DTs) are a type of supervised ML technique for classification that uses a set of if-then rules to simplify the process and improve human comprehension. The two types of nodes in a decision tree are the leaf nodes, which represent the outcomes, and the decision nodes, which represent the choices that lead to those outcomes (choice between alternatives). A decision tree can be used to predict a class or target by inferring decision rules from training data. The decision tree has the benefits of being easy to understand, helping eliminate confusion while making choices, and facilitating in-depth research. Connectivity, anomaly detection, data aggregation, and mobile sink path selection are just a few problems that WSNs can address with the help of adopted decision trees.

The algorithm employs a divide-and-conquer technique. This algorithm, in contrast to ID3, incorporates normalizing procedures. The algorithm determines the ratio based on the values of the information acquired. Building and repositioning intermediate trees is feasible at the time of the decision tree's inception. The decision tree method also employs branch pruning to eliminate potentially incorrect data and lower the error rate. A single node must be established for the tree-building method to begin; if all of the samples belong to the same class, processing will continue; otherwise, the node will be established as a leaf and will not represent any classes. An optimal segmentation attribute is chosen if a node has characteristics from multiple classes, and the tree expands from there.

Each feature's information gain is computed, and the feature with the highest value is chosen as the tree's decision node. During the election of the cluster leader, this is the best time to identify and remove any malicious nodes. After a decision node has been identified, the procedure continues by creating a child branch off of that node. If all the elements in the subgroups listed above have the same value, then the procedure ends, and that value is used as the output. The process ends if the subgroup contains exactly one node and no distinguishing features are identified.

3.6.3 K-Means Clustering

With minimal effort, the k -means method can divide a data set into a specified number of groups. Each remaining point has its nearest center determined by randomly selecting k sites. After the data is partitioned into clusters, the centroid of each cluster is recalculated. Each time the algorithm is run, the cluster's centroid shifts until the algorithm reaches a plateau and no cluster centroid shifts. Suppose we define n as the total number of points and k as the total number of centroids. In that case, i is the total number of iterations, and d is the total number of attributes, then we can say that the time complexity of the k -means algorithm is O (n*k*i *d). In equations (3.4). We see the minimization function for the sum of squares of errors:

$$\min f(x) = \sum_{i=1}^{k}\sum_{j=1}^{N}\left\| x_i - y_j \right\|^2 \tag{3.4}$$

Where N is the number of data points in the ith cluster and $\left\| x_i - y_j \right\|$ is the Euclidean distance between x_i and y_j. The simplest clustering method, k-means, is also useful in WSNs for identifying ideal cluster heads (CHs) and detecting of malicious nodes to employ while transmitting data to the base station. This method also works well for locating productive mobile sink rendezvous spots. Choosing a different value of K can affect the outcomes in some situations. Getting the best results from the analyzed data is crucial to get the value of k right. Euclidean, Manhattan, and Minkowski are just a few of the distance and neighbour node formulas that can be applied. Here are the relevant formulas as in (3.5).

$$\begin{aligned} &Euclidean \rightarrow \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \\ &Manhant\tan \rightarrow \sum_{i=1}^{k}\left| x_i - y_i \right| \\ &Minkowski \rightarrow \left[\sum_{i=1}^{k}(\left| x_i - y_i \right|^q) \right]^{1/q} \end{aligned} \tag{3.5}$$

Each data point has its own set of attributes.

3.6.4 Hybrid-Ensemble Machine Learning

Combining multiple machine learning algorithms into an ensemble makes the classification more accurate and faster. This approach involves several learning procedures using various machine-learning approaches and then combining and categorizing the results. The underlying algorithm performs two basic steps. At first, the original dataset is partitioned, and the distribution of a basic model is generated on those subsets. After doing so, the distribution is aggregated into a single model, and the results are obtained. The stacking strategy differs from standard machine learning methods because it involves a model production step. Models built from the training set are combined. You can describe the algorithm's function as follows:

- Models are created during training by employing the dataset and the training method.
- Each derived model has full annotations for all the dataset's training samples.

- The final model is built from the other models in the training dataset using the combiner method.
- After a final model has been obtained, it is used to categorize test dataset samples.
- A final prediction is made using the final model once all test dataset samples have been classified and the class predicted by the stacking algorithm of the sample is chosen.

The term ensemble technique is used to describe three distinct approaches. We're bagging, boosting, and stacking here. Data mining approaches and the capabilities of the combiner models used by each of these methods are where they diverge. As opposed to maximizing predictive power like boosting does and minimizing variance as bagging does, stacking strives to do both. The function that generates a single model uses the average weight in the bagging strategy, the weighted majority vote in the boosting approach, and Logistic regression in the stacking approach.

The clustering technique divides the dataset into groups with various dimensions making new discrete features. The new clustering scheme uses the Gaussian mixture model (GMM) to develop new features [144], as shown in Figure 3. 9. The GMM method is a clustering model without any training requirement. After adding and allocating additional features, the dataset is modified for better performance and accuracy.

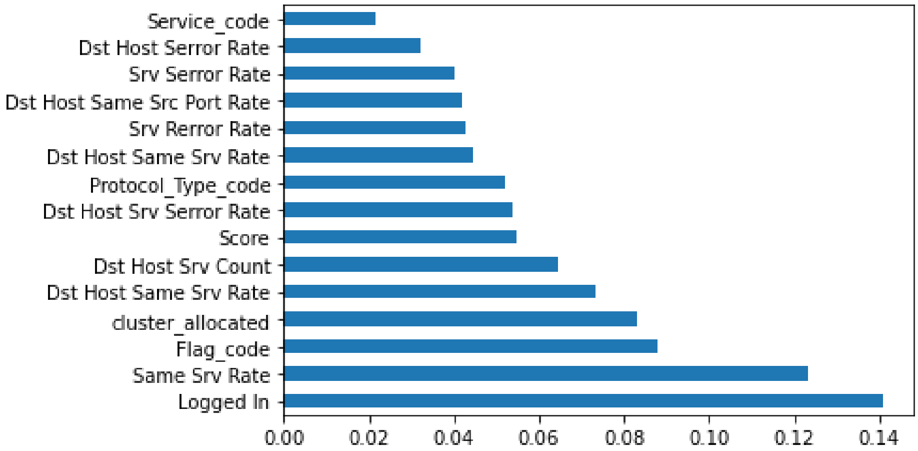


Figure 3. 9. Input extraction feature after modified NSL-KDD using Gaussian mixture clustering.

The new hybrid technique uses features and information using the clustering technique to the data set. The proposed scheme uses ten-fold cross-validation using 80% of training and 20% of testing using the benchmark dataset for attack detection and classification.

3.6.4.1 Gradient Boost

Extreme Gradient Boosting (XGBoost) is a fast and effective classification method for massive datasets [130]. With less memory required for training and testing the dataset for classification, the Gradient Boosting method is able to increase computational performance and produce accurate results for intrusion detection [145]. As demonstrated in equation (3.6), it is a powerful machine learning strategy for maximizing the loss function and computed features.

$$\Phi\left(X\right)=\sum_{k=1}^{K} f_k(X) \quad \mathrm{f}_k \varepsilon F \tag{3.6}$$

Where $\phi(\mathrm{X})$ is the final result of the K sequential classifier and f_k, is the decision tree for the K number of iterations in the Gradient descent algorithm. The method additionally employs parallel computing to categorize the required results. To optimize the process and control the overfitting factor, XGBoosting enhances the Gradient descent and regularization approach. Classifier parameters are connected using the equation (3.7) shown below:

$$\ell\left(\phi\right)_t=\sum L\left(f_{t-1}-f_t\right)+\Omega\left(f_t\right) \tag{3.7}$$

Where $\mathcal{L}(\phi)_t$, is the loss function, and $\Omega(f_t)$ is the regularization term to optimize the step size t. When retrieving scores for an attribute, the Gradient Boosting method employs feature metrics to determine how the attribute was measured.

3.6.4.2 Ensemble Learning

Ensemble machine learning algorithms use classifiers with averaged accuracy to mitigate the potential for overfitting and bias introduced by a single classifier [146]. Tree-based models used in machine learning for classification improve precision. Ensemble methods are meta-algorithms that aggregate separate machine learning prediction models for assessing stacking and variation [147]. In order to boost overall performance, the Ensemble method combines multiple machine learning outputs into a single, more resilient model. The fundamental ideas behind ensemble techniques are stacking, bagging, and boosting. Through stacking, we may combine and strengthen the predictive ability of multiple machine learning models.

To further improve the classification of the hybrid machine learning models for the proposed scheme on the benchmark dataset, a tree based on the Parzen estimation (PTE) is used in conjunction with hyperparameter and Bayesian optimization (BO) techniques. Hyperparameters are used in every machine learning task to fine-tune these parameters and achieve the best possible outcomes. As a result of this hyperparameter optimization (HPO), the efficiency and effectiveness of machine learning can be increased with less human input [148]. Hyperparameter optimization is also used in the black box and global optimization for more accurate function evaluation. As a result, we can explain the inner workings of Bayesian Optimization without getting too technical. In HPO for deep neural networks, Bayesian optimization (BO) is gaining popularity as a framework for global optimization with costly black-box functions. Bayesian optimization is a recursive method that uses a probabilistic surrogate model and an acquisition function to evaluate choices with the help of the Gaussian process. Random forest and tree Parzen estimators are just two of the tree-based approaches used to deal with hyperparameters (PTE). This suggested effort combines Bayesian-based optimization (BBO) with tree Parzen estimators (TPE) to determine the optimal evaluation point for fully automated machine learning.

3.6.5 Performance Metrics

Detection rate [18], precision, false-negative rate, and the receiver operating characteristics curve (ROC) are the performance parameters that may be measured and analyzed. These indicators evaluate the system's performance, generate a categorization report, and compare the results to those of other studies. We used a complexity matrix as one of our criteria for rating submissions [140]. Values from the complexity matrix are used to determine the criterion for evaluation. Following is a breakdown of the values in the complexity matrix:

- In the dataset, TP (true-positive) refers to the number of samples accurately predicted to be incursions.
- The number of samples in the normal class that were correctly predicted to be in the normal class (true-negative, or TN).
- False-negative (FN): The percentage of abnormal samples incorrectly labelled intrusions. The number of normal samples in the dataset was wrongly classified as incursions (FP or false positive).

The detection rate is calculated by dividing the TP value by the total number of samples for which intrusion estimates were calculated. The accuracy value measures how well a system performs in classifying data by comparing the fraction of data points correctly labelled by the system to the total number of data points. To demonstrate the system's efficiency, we employ the following mathematical equations (3.8)-(3.9):

$$\text{Detection Rate} = \frac{TP}{TP+FN} \tag{3.8}$$

$$\text{False alarm rate} = \frac{FP}{TN+FP}$$

$$\text{Precision(PR)}=\frac{TP}{TP+FP}$$

$$\text{Recall(Rc)}=\frac{TP}{TP+FN}$$

$$\text{Accuracy} = \frac{TN+TP}{TN+TP+FN+FP} \tag{3.9}$$

$$\text{F1-score} = \frac{2\times RC\times PR}{RC+PR}$$

It has been noted that the designed IDS can perform four different outcomes for each traffic operation. The following scenarios are generated using the confusion matrix: First, a True Positive (TP) occurs when an intrusion detection system (IDS) reports a successful detection of malicious activity on a network [149]; second, a True Negative (TN) occurs when an IDS does not report a successful detection of malicious activity, third, a False Positive (FP) occurs when an IDS reports no malicious activity, and fourth, a False Negative (FN) occurs when an IDS reports a successful detection.

Mean squared error (MSE): Mean squared error (MSE) measures the amount of error in statistical machine learning models for computing the position and distance of the wormhole attack between two points. It assesses the average squared difference between the observed and predicted values of each sensor node's position and location, having its unique identity to detect routing attacks. When a model has no error, the MSE equals zero. As model error increases, its value increases. The mean squared error is also known as the mean squared deviation (MSD) defined by the equation (3.10).

$$\text{MSE}=\frac{\sum_{i=0}^{n}(x_i-\overline{x}_i)^2+(y_i-\overline{y}_i)^2}{n} \tag{3.10}$$

Where:

- x_i and y_i is the ith observed values.
- $\overline{x}_i$ and $\overline{y}_i$ are the corresponding predicted values.
- n is the number of observations.

The mean squared error uses a formula quite close to the variances. The MSE is calculated by square-rooting the difference between the observed and anticipated values. That should be done for every observation. After that, divide the total by the total number of observations to get the square root.

Training Time: Training time measures the time required to train or build the system. It can be an essential metric, especially when efficiency or real-time performance is crucial.

3.7 Simulation and Environmental Setup

Network design and model simulations were executed in MATLAB R2021a on a Windows 10 64-bit x64-based processor running an Intel Xeon Silver 4214 CPU at 2.20GHz 2.19GHz (2 processors), with 128GB (128GB useable) of installed RAM. Data processing and analysis with machine learning classifiers are performed in Python libraries, including Keras, numpy, Sklearn, Seaborn, and pandas, using Anaconda Navigator and MATLAB R2021a. The simulation parameters for running network attack scenarios are depicted in Table 3. 5, along with the values. This study assumes that Node-0 is the final destination for network traffic. A total of 5 seconds are allotted for the simulation. Wormhole routing attack simulations are run, with results generated by an artificial neural network and genetic algorithm that have been genetically optimized for maximum efficiency. Next, a malicious node is introduced to generate and extract features for both benign and malicious network traffic, and this process is repeated for another 5 seconds of simulation time to develop a new database.

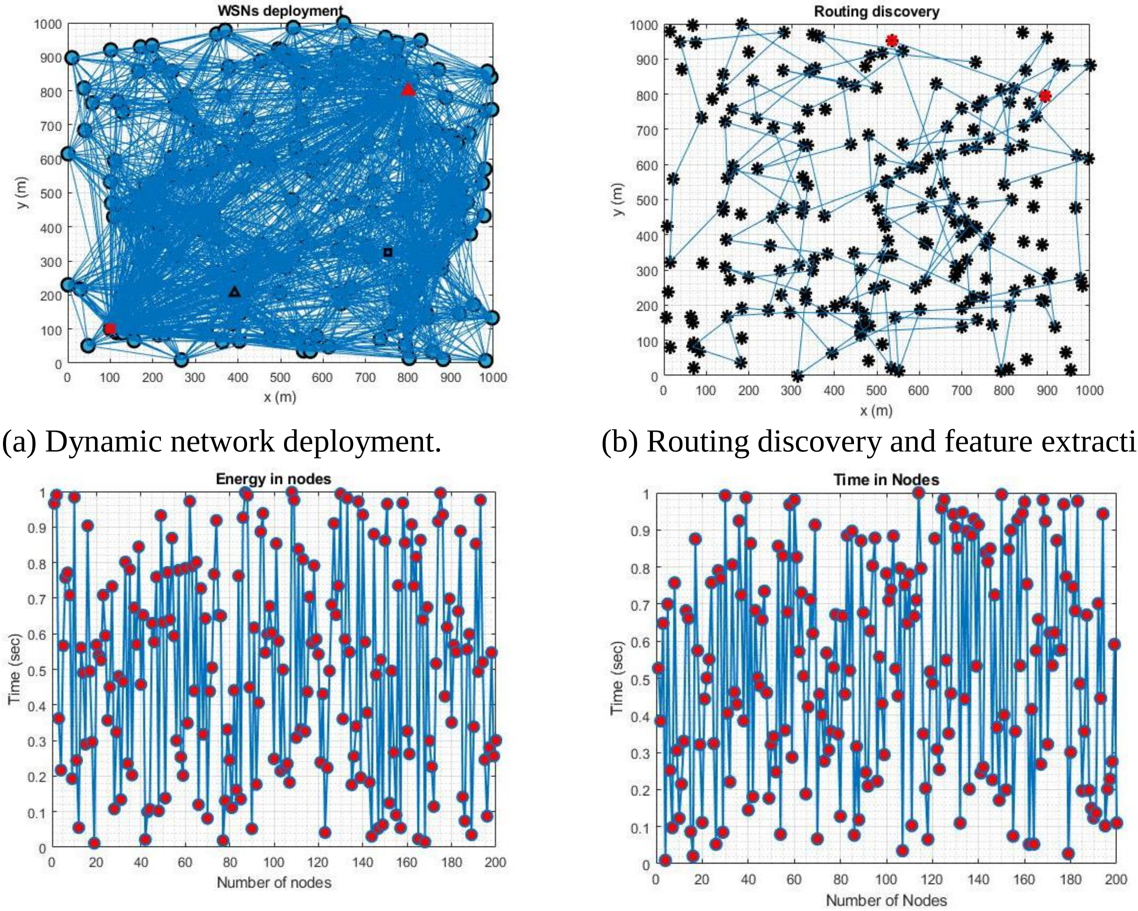
Table 3. 5. WSN configuration of Simulation setting.

Parameter	Setting	Parameter	Setting
Base Station	1	Topology	Hierarchical
Filed size	1000 ×1000 m^2	Number of attacks	2
Number of Nodes	200	Mobility Model	Random
Protocol type	Routing	Number layers	10
Cluster size	10	Max epochs	200
Attack type	wormhole	Data size	5000 Kb
Number iterations	200	Simulation Time	5 Seconds

Table 3. 5 displays the simulation scenarios that use a mobility and routing protocol based on a random selection of mobility and intermediate nodes to discover and extract features from potential routes. A wormhole attack is injected between two malicious nodes and creates a tunnel.

3.7.1 Experimental Results and Analysis

In this section, the wireless sensor networks are dynamically fully connected, forming wormhole tunnels for routing discovery and detection of routing attacks as in Figure 3. 10 (a) and (b). The routing wormhole attack highly affects the sensor nodes' energy consumption and timing operation for effective communication, as shown in Figure 3. 10 (c) and (d).



(a) Dynamic network deployment.

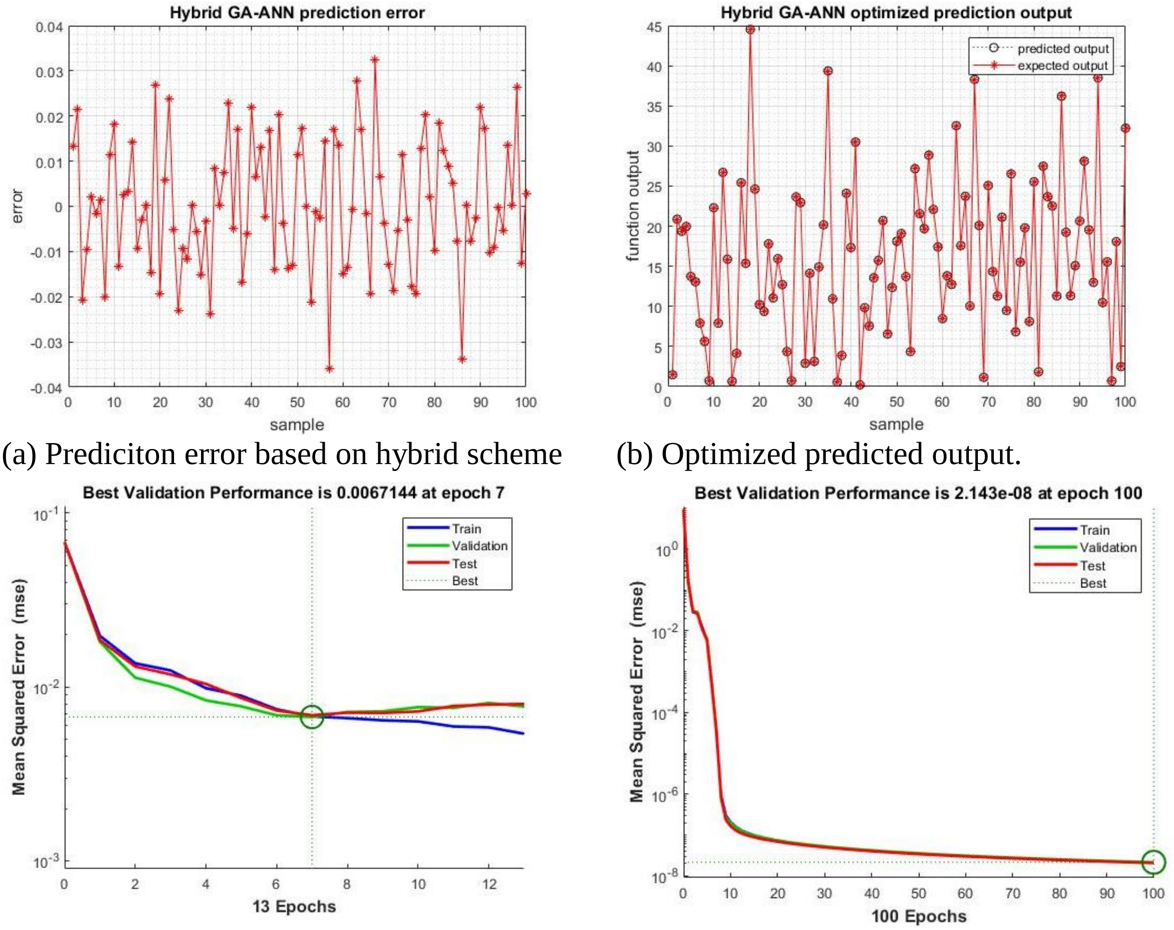
(b) Routing discovery and feature extraction.

(C) Energy consumed for nodes with time.

(d) Time elapsed for each node.

Figure 3. 10. wireless sensor networks deployment and routing discovery concerning energy consumption and time consumption.

The simulation results show that the proposed attack detection and classification techniques are effective, with an average detection accuracy of 99.46% by varying the hope count and wormhole tunnel of the routing attacks across the network. The results also show that hybrid techniques improve the prediction error and maximize the performance, as shown in Figure 3. 11 (a) and (b). The validation and performance of the proposed system are effective for the detection and localization of routing attacks, as shown in Figure 3. 11 (c) and (d) at epochs 7 and 200 epochs with minimum mean squared error (MSE) of 0.0067 and $2.143x10e^{-08}$.



(a) Prediciton error based on hybrid scheme (b) Optimized predicted output.

(c) Best performance ate epoch of 7 (d) Best performance at the epoch of 100.

Figure 3. 11. Performance evaluation of the proposed system using hybrid GA-ANN taking 100 samples and varying the epochs.

3.7.2 Intrusion Detection Analysis

The samples from the reference datasets have been put through training and testing processes [150]. First, we randomly assign each sample to two groups: the training and test sets. Step two involves using the whole training set for both training and testing. Finally, cross-validation was utilized to test how well the proposed model worked. The area under the curve, false alarm rate, precision, and classification accuracy are used to evaluate performance. The performance of the proposed technique is evaluated using machine learning models using a benchmark dataset containing a class of attacks in wireless sensor networks. The hybrid optimized machine learning also utilizes the same benchmark dataset to evaluate the proposed routing attack localization and detection effectiveness in wireless sensor networks. Table 7 shows the comparative performance of the various schemes of machine learning. The performance of the proposed system is further improved with the application of cluster labelling (CL) k-means binary classification techniques. Table 3. 6 shows the comparative performance of the various hybrid machine learning techniques.

Table 3. 6. Comparison performance of hybrid machine learning models using the benchmark datasets.

Classifier	Results against the UNSW_NB15				Results against the CICIDS2017 dataset			
	Accuracy	Precision	Recall	F1core	Accuracy	Precision	Recall	F1Score
XGB	99.78	99.74	99.78	99.75	99.82	99.86	99.82	99.83
RF	99.75	99.67	99.75	99.70	99.82	99.82	99.82	99.80
DT	99.68	99.63	99.68	99.66	99.82	99.91	99.82	99.85
ET	99.72	99.66	99.72	99.68	99.82	99.80	99.82	99.80
ES	99.78	99.74	99.78	99.75	99.82	99.91	99.82	99.85
CLK-M	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

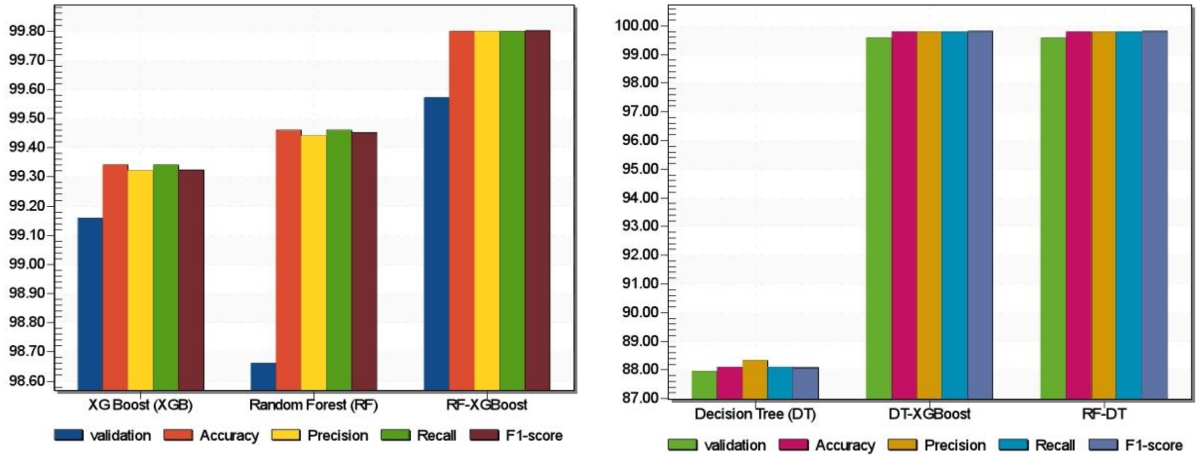
The hyperparameter and Bayesian optimization (BO) techniques and the tree-based Parzen estimation (BO-PTE) are used to boost the performance of hybrid machine learning models for the proposed system. The performance of the proposed scheme is also evaluated using a UNSW_NB15 benchmark dataset with the application of various machine learning models, as shown in Table 3. 6. The binary classification technique using hybrid cluster labelling K-means achieves better classification accuracy of 100% using the benchmark dataset.

Table 3. 7 shows how merging different hybrid machine learning models and moving data frames from one machine learning to another improves the proposed system's performance even further.

Table 3. 7. Comparison of various hybrid machine learning models using the NSL-KDD dataset.

ML Classifiers	Performance evaluation metrics					
	validation	Accuracy	Precision	Recall	F1-score	T. time
NB	83.94	83.71	90.35	83.71	85.68	0.025
DT	87.94	88.10	88.34	88.10	88.07	0.22
XGB	99.16	99.34	99.32	99.34	99.32	1.83
RF	98.66	99.46	99.44	99.46	99.45	0.17
DT-XGB	99.57	99.80	99.80	99.80	99.80	1.24
RF-XGB	99.57	99.80	99.80	99.80	99.80	1.44
RF-DT	99.57	99.79	99.79	99.79	99.79	0.327

The results demonstrate that hybrid ML models outperform their standalone counterparts in terms of Validation, Accuracy, Precision, Recall, F1 score and training time. When it comes to classification and detection, based on the results and model validation, we can say that the created system has high classification and detection accuracy against DoS attacks in WSNs.The results show that hybrid machine techniques perform better attack detection and classification of attacks using the NSL-KDD benchmark dataset, as shown in Figure 3. 12 (a) and (b). For attack detection and classification, the extreme Gradient boosting (XGB)-an enhanced hybrid of a random forest and a decision tree achieves better results than either the random forest or the decision tree alone in terms of validity, accuracy, precision, recall, and f1-score.

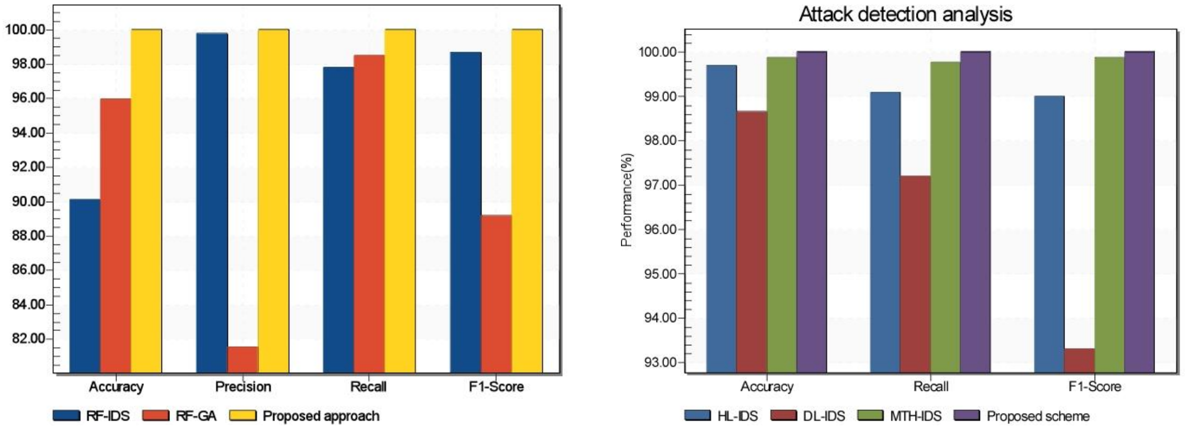


(a) RF-based comparative analysis (b) DT-based comparative analysis

Figure 3. 12. Performance comparison of various machine learning models using NSL-KDD.

The performance of the proposed technique is effective compared to L. Yang et al. [41] developed multi-tiered hybrid intrusion detection systems (MTH-IDS) for secure vehicular networks using the benchmark dataset CICIDS2017 for known and unknown attacks and achieved average detection accuracy of 99.88% using binary classification. P. Sun et al. [51] developed a hybrid deep learning-based intrusion detection system (DL-IDS) using a convolutional neural network and a long short-term memory network (CNN-LSTM). The scheme achieved an average detection accuracy of 98.67% by extracting the network traffic. This proves that the proposed scheme effectively detects DoS attacks using the benchmark dataset in wireless sensor networks, as shown in Figure 3. 13 (b), using attack detection performance metrics. S. M. Kasongo [151] presented an intrusion detection system for the Internet of things using random forest based on a genetic algorithm (RF-GA) for feature selection, as shown in Figure 3. 13. This achieved average detection accuracy of 87.61%, which is less than compared to 100% using hybrid binary classification. M. F. Suleiman and B. Issac [152] Evaluated six machine learning classifiers using UNSW_NB15, phishing and NSL-KD benchmark datasets for intrusion detection system. Random forest based intrusion detection system (RF-IDS) produced better detection accuracy using UNSW_NB15. Temporal and spatial features to enhance attack detection and classification. This confirms the proposed technique is effective for detection and localization of attacks as shown in Figure 3. 13 (a).

In order to achieve high-performance intrusion detection across a wide range of attack types, B. Media et al. [140] proposed a hybrid-layered IDS (HL-IDS) that employs a number of distinct machine learning and feature selection approaches, as shown in Figure 3. 13 (b). The size of the NSL-KDD dataset is decreased in the created system by first performing data pre-processing on the dataset using various feature selection algorithms.

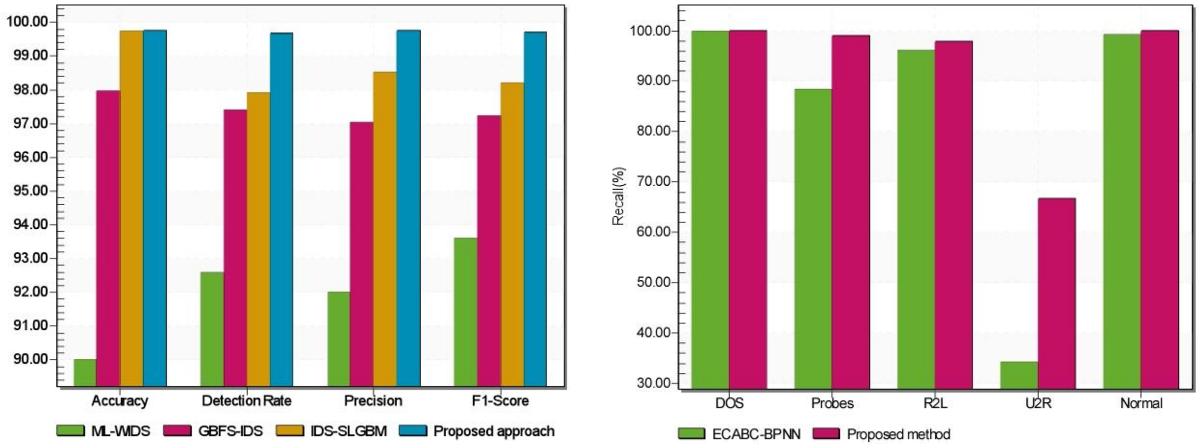


(a) Comparison based on UNSW_NB15 (b) Comparison-based CICIDS2017

Figure 3. 13. Performance comparison of the proposed scheme-based hybrid machine learning techniques using benchmark datasets.

G. H. Lai [153] Proposed the detection of wormhole attacks in wireless sensor networks using low power and lossy network (LLN) routing protocol and achieving 100% accuracy with fixed range and wormhole tunnel points. This confirms the proposed technique is effective for localizing and detecting routing attacks in wireless sensor networks using a benchmark dataset. Y. Yuan et al. [154] Presented a novel lightweight method for Sybil attack detection in distributed WSNs using the approximate point-in-triangle (APIT) localization approach. They achieved an average detection rate of 90%, which is less than the proposed work. D. Upadhyay et al. [145] proposed a framework for intrusion detection systems in smart grids using Gradient boosting feature selection by applying machine learning classification techniques. The scheme combines feature engineering with machine learning classifiers and achieves the performance in Figure 3. 14 (a). This suggests that the proposed method is effective for DoS attacks in wireless sensor networks in various applications.

A unique feature selection algorithm, the dynamic recursive feature selection algorithm, was introduced by Nancy P et al. [155], which chooses an optimal number of features from the data set. What's more, there's a sophisticated intrusion-detection system based on a fuzzy logic algorithm (IF-IDS) using the NSL-KDD dataset. Extending the decision tree approach and including convolution neural networks are also presented as means by which to detect the invaders efficiently. The technique of intelligent feature selection algorithm named dynamic recursive feature selection algorithm (DRFSA) has been proposed in this work, which picks the important features to construct the data set. G. Qi, J. Zhou et al. [156] presented a new ECABC-BPNN, a combination of back propagation neural networks (BPNNs) and elite clone artificial bee colonies (ECABCs), that improves upon the standard BPNN's weight and threshold settings as shown in Figure 3. 14 (b).



(a) Comparative analysis of the proposed scheme

(b) Comparative analysis based on recall

Figure 3. 14. Performance comparison of the proposed technique with previous works.

The next step is using ECABC-BPNN to identify threats in a computer system's network. The comparison and Conducted experiments on assault classification using benchmark dataset as shown in Table 3. 8.

Table 3. 8. Performance evaluation of the proposed system using the NSL-KDD dataset.

	Proposed AIDS-HML approach			Intelligent fuzzy-based IDS			ECABC-BPNN
Attacks	Precision	Recall	F1Score	Precision	Recall	F1Score	Precision
DOS	99.89	99.93	99.91	99.99	97.23	98.72	98.56
Probes	99.30	98.87	99.08	92.67	97.97	93.34	86.70
R2L	98.91	97.84	98.38	57.39	28.32	42.97	97.20
U2R	80.00	66.66	72.72	95.23	63.56	59.03	83.67

The proposed scheme is further compared for validation with previous works to detect and localize routing attacks in WSNs. S. Jiang et al. [157] Proposed an intrusion detection system based on a secure light gradient boosting machine (IDS-SLGBM) in wireless sensor networks using the WSN-DS benchmark dataset with the class of routing attacks shown in Figure 3. 14(a).

3.8 Conclusion and Future Work

The proposed advanced intrusion detection system based on machine learning effectively detects and classifies attacks for scalable and manageable in hierarchically distributed wireless sensor networks. This research aims to create a classification model for an advanced intrusion detection system based on hybrid machine learning, specifically tailored for use in wireless sensor networks to detect intrusions. Each sensor node collects information on the state of its features and reports it to the cluster's central processing node. The cluster leader checks the data and then forwards it to the main cluster head. The proposed hybrid machine learning models use training and testing data to identify attacks. Our suggested IDS-HML outperforms state-of-the-art systems regarding detection and localization accuracy in a simulated attack on a WSN. Comparing the hypothetical outcomes to earlier research shows that they are credible. The simulation results show that the proposed system is effective for detecting routing attacks with a localization accuracy of 99.46% of the wormhole routing attacks. The effectiveness of the suggested system has been measured in accuracy, precision, TP Rate, FP Rate, F-Measure, Mean squared error, and Time. The designed IDS-HML achieved 99.82%, 99.91%, 99.85%, 99.82%, and 100% for average detection accuracy, precision, F1-score, recall, and CLK-Means respectively, in the presence of normal and intrusion traffic using CICIDS2017 dataset as a benchmark for multiclass and binary classifications. This model uses logic rules for decision-making and interpretable predictive models.

Although the proposed method performs well, it is essential to note that IoT-based WSNs are still susceptible to sinkholes, Sybil attacks, blackhole attacks, selective forwarding, and classification attacks not addressed in this study. The countermeasures module is only provided in concept, which is another shortcoming. Therefore, we plan to investigate and eventually offer specialized advanced hybrid intrusion detection systems for each type of assault utilizing benchmark datasets to evaluate hybrid machine learning techniques. In future work, we will explore collaborative advanced intrusion detection systems based on machine learning in IoT-based wireless sensor networks for different applications using benchmark datasets for evaluations.

Chapter 4

4 DESIGN AND DEVELOPMENT OF SECURE LOCALIZATION FRAMEWORK

The main goal of this research objective, Secure Localization Techniques, is to develop methods and algorithms that allow secure and accurate localization of wireless sensor nodes. High lights of this objective are:

- Developing a framework for secure and precise wireless sensor nodes and localization to detect malicious nodes.
- To mitigate security risks associated with localization techniques in IoT-WSNs.
- Establish secure communication protocols for the localization of wireless sensor networks.
- Design and develop methods for localizing wireless devices to reduce power consumption and improve network lifetime.
- Explore privacy-preserving approaches to wireless localization with reduced latency and communication overhead.

4.1 Introduction

Data collected by wireless sensor nodes is sent to the cluster's hub, where it is aggregated. An intelligent sensing and computing framework is required for security purposes, such as the localization and detection of threats using an artificial neural network (ANN). This method is gaining popularity because of its low computing cost and fast convergence. In this chapter, we proposed a multilayer perceptron artificial neural network to identify and locate routing attacks in WSNs. Both traffic management and object tracking are promising new uses for WSNs, and both rely on pinpointing where individual sensor nodes are [196] precisely. It is critical to have a reliable prediction of the sensor node's location for effective routing and location-aware services. WSN data is sometimes worthless unless the location of the sensor is known. Researchers are increasingly interested in localization methods due to their potential usefulness in various WSN applications. Localization methods can be categorized as range-based methods, which use known lengths or angles between nodes to create position estimates, and range-free methods, which use the proximity of a set of reference nodes to make location estimates. Because of their cheaper hardware and computational requirements, range-free methodologies are gradually replacing range-based methods in WSN localization. As an example of a range-free technique, the clustering and localization algorithm is well-known since it estimates a node's position based on the positions of surrounding reference nodes. The initial placements of the reference nodes are either hard-coded or calculated during the setup process.

4.1.1 Routing Attacks

If an authentic node is rendered unreachable by an attack on the network layer of a wireless sensor network, the network as a whole will suffer. What follows is a discussion of the most frequent forms of attack against WSNs. Information security is breached before arriving at the target node with these kinds of assaults. Wireless sensor networks are particularly vulnerable to assaults because they can be compromised at any of the many levels inside the network.

4.1.1.1 Wormhole Attack

Two hostile nodes with a common goal and a tunnel connecting the two locations can launch a wormhole assault. A wormhole attack can lure and redirect large network traffic to conduct multiple malicious attacks. It uses the intermediate nodes to broadcast its packets, which can then be sniffed, modified, and dropped [28]. Figure 4. 1 shows when the packet is sent from the source node S to the destination node D, as shown in cluster A; this is an example of a wormhole assault. Between A and B, the malicious nodes, a wormhole is formed. Before reaching the base station, the tunnel modifies the packet and causes it to be dropped. At least a wormhole attack can be detected by the two hostile nodes using a tunnel's secure communication channel [158]. The wormhole connection will now begin to receive data packets and forward them onward. A control packet is sent to the malicious node at the other end of the tunnel. Finally, it forwards the packet through a secure connection to another node that has captured its attention. The private channel is chosen as the means of transmission between the source and the destination when more desirable metrics are to be achieved, such as a shorter total journey time or a reduced number of hops. There are often two parts to the attack. Each wormhole node may be interested in several different starting directions. In the second phase, the packets start utilizing the compromised nodes. The network's efficiency may be hampered in many ways by these nodes. Nodes in the wormhole can be utilized for theft if they are able to delete, alter, or transmit data without detection.

4.1.1.2 Sinkhole Attack

The sinkhole node acts as if it were a normal node and sends out routing advertisements to the base stations, leading the other nodes in the network into a false sense of security. The malicious nodes puncture the routing path through their malicious actions, which might disrupt the network's normal functioning. In a sinkhole attack, a hacked node closer to the destination promotes the route to the attackers. Through this manipulation of routing data, the genuine node is misled and drawn closer, as in Figure 4. 1. The sinkhole attack scenario is depicted by cluster B, which actively seeks out and captures packets from neighbouring nodes. Sinkhole attacks use a hidden tunnel to lure in nodes and steal data packets. Then, the rogue node tricked the base station into receiving packets it had not originated.

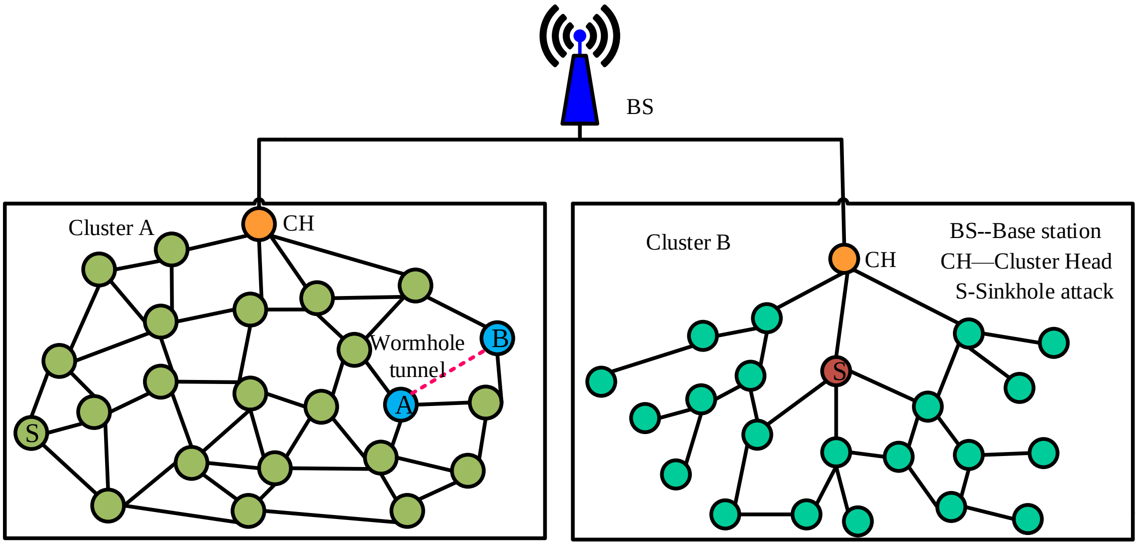


Figure 4. 1. Wormhole attacks and sinkhole attacks on WSNs are depicted in Cluster A and B, respectively.

4.1.1.3 Blackhole Attack

Capturing sensor nodes and reprogramming them to reject packets rather than receive and forward them to the base station is the essence of a blackhole attack [28]. When a malicious node enters the black hole region, it compromises the information with it. By dividing the network this way, the black hole attack prevents vital updates from reaching the base station, degrading the network's performance. It harms network performance indicators and uses up a lot of bandwidth. As demonstrated, a packet is sent from the source node S to the base station via the intermediate nodes C and D, as in Figure 4. 2 of cluster 1. Every packet is eaten by the blackhole node instead of being forwarded to its intended receiver.

Black hole attacks use security weaknesses to track down a network's routing information and carry out malicious actions [27]. The black hole attack prevents the packets from reaching their destinations because the packets are being dropped. The suspicious node causes packet drops for selected nodes and allows for granular control over which nodes receive these drops. The black hole discards incoming data and relays spoofed packets back to the base station. In comparison to a typical node's request and answer, the higher order number of black hole attacks results in routing requests and responses with a higher total amount of bits. To prevent the typical node's reply to a routing request with a higher-than-normal order number, thus preventing routine network elimination [159].

4.1.1.4 Sybil Attack

Regarding wireless sensor networks, a Sybil attack can fake out other nodes by pretending to be the real deal [29]. The routing table and the trustworthiness of the legitimate node are both disrupted by a Sybil assault. Sybil attacks use fake identities to trick nearby nodes into thinking they're in danger [160]. Authorized nodes are the focus of this assault, which employs geographic routing methods. Sybil attacks use multiple fake identities to conceal their legitimate node's data [161], as shown in Figure 4. 2 of cluster 2. Through multiple transmissions, the malicious node includes fictitious events. As this attack relies on deception, it isn't easy to detect the network as a whole.

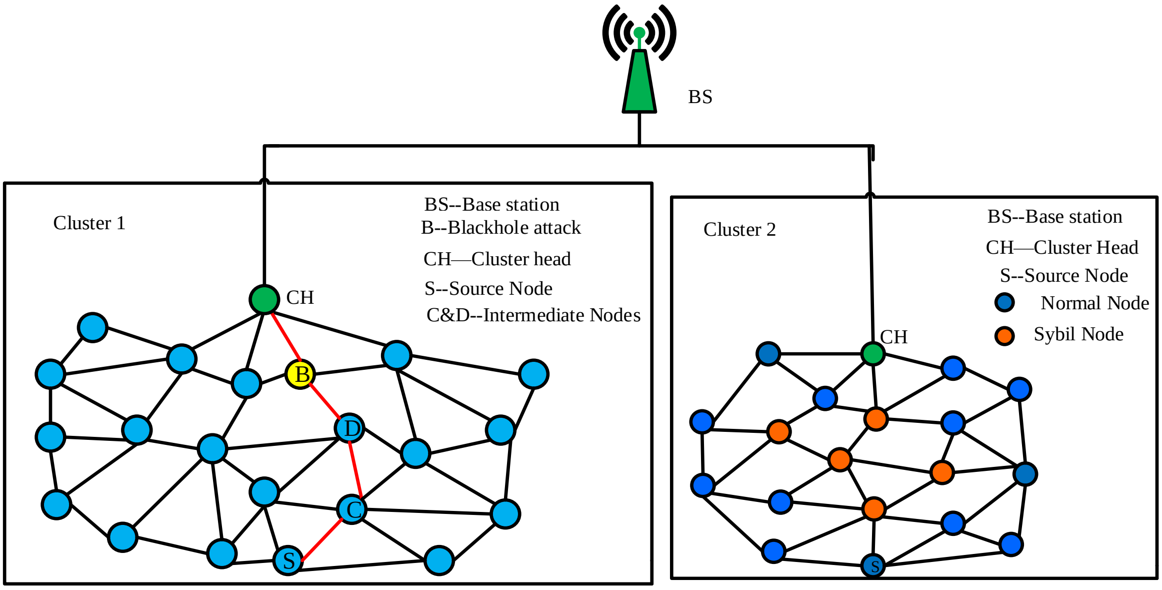


Figure 4. 2. Clusters 1 and 2 illustrate black hole and Sybil assaults scenarios in wireless sensor networks (WSNs), respectively.

4.2 Problem Statement

The problem addressed in this chapter is the need for optimal and intelligent localization methods for accurate node position and attack identification in Wireless Sensor Networks (WSNs). The existing literature lacks comprehensive studies that consider multiple attacks, localization accuracy, and detection efficiency. To overcome this problem, we proposed a security localization and detection scheme called DV-Hop-RSSI-DE, which combines hybrid localization approaches with a multilayer perceptron artificial neural network. The goals of this chapter are:

- By employing the DV-Hop-RSSI-DE method, the scheme aims to improve the accuracy of node localization and enhance the detection and classification of attacks in WSNs.
- The chapter also discusses the design and planning of a distributed hierarchical clustered topology, which includes sink nodes, cluster heads, malicious nodes, and sensor nodes.
- To evaluate the effectiveness of the system, the method utilizes benchmark datasets such as CICIDS2018, UNSW-NB 15, WSN-DS, and NSL-KDD. These datasets are used to train and test the proposed system, and various evaluation metrics are employed to assess its performance.

4.3 Research Contribution

A multilayer perceptron neural network optimized for this purpose is the foundation of the proposed assault localization and detection technique [162]. Phases of the proposed system are implemented after careful consideration has been given to network design and node configuration. Network attack detection and classification processes involve data processing, feature extraction, training, and testing. This work makes some new contributions, including:

(1). The primary goal of this project is to develop and simulate a topology for a wireless sensor network that can detect and localize attacks.

(2). As a means of learning about and practising countermeasures against routing assaults, using clustering and routing protocols as a model is recommended.

(3). Use attack detection localization metrics to evaluate the network using a representative public dataset as a reference point.

(4). Investigate the use of machine learning to secure localization and route detection in wireless sensor networks across all network layers.

(5). By analyzing network traffic data and extracting relevant features, hostile nodes can be identified and classified using a multilayer perceptron neural network approach. Finding the suspicious node with the greatest precision is achieved.

(6). To detect and pinpoint numerous attacks with improved classification accuracy for clustered and hierarchical network architecture.

(7). Use a comparative performance metric to validate and certify the scheme's security against similar prior works.

(8). Explore hybrid range-based and range-free localization techniques for unknown and malicious nodes that affect the quality of service in WSN using a collaborative approach.

4.4 Hierarchical Network Model

The simulated network includes symbols for a sink, a cluster head, a sensor, and an adversary node as in Figure 4. 3.. The sensor nodes cluster in predictable patterns across the system. Each cluster selects a single node to serve as its hub for relaying data to the other nodes in the cluster, the beacons, and the base station. The beacon nodes optimize their communication paths with the rest of the network using a fitness function. Beacon nodes and anchor nodes mean the same thing here. You can utilize attacks like Sybil and wormhole assaults on this network paradigm. Anchor nodes are located and positioned about other nodes. An anchor node's position in a network is always fixed once established there. The network of sensors has staked its claim. The localization strategy is used for pinpointing the precise location of each sensor node by grouping them into cohesive units led by a single node [163]- [164].

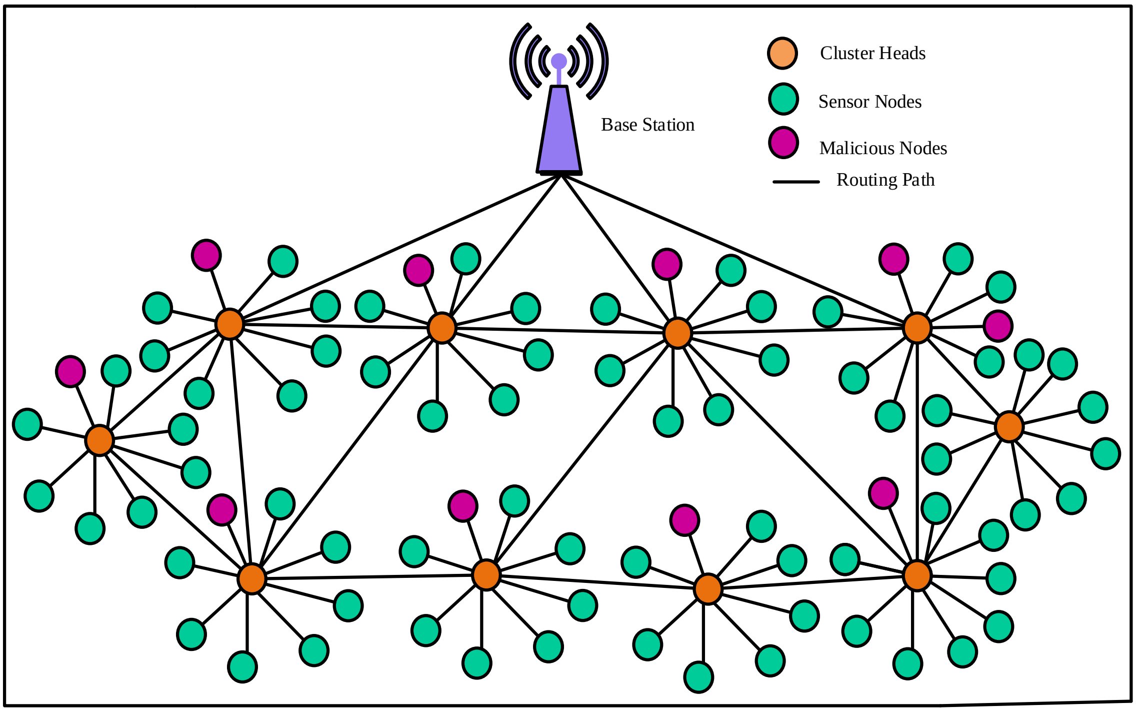


Figure 4. 3. Clustering in hierarchically distributed wireless sensor networks model.

When a node's position is unknown, it can use the help of the anchor node to figure out where it is. The system periodically transmits updated locations for the sensor nodes. By assuming several fictitious identities, malicious nodes can advertise their locations as though they were the network's official beacons. Meanwhile, the malicious node is busy constructing tunnels to discard data before reaching its intended destination. To reduce power consumption and increase network lifetime, hierarchical clustering of sensor deployments is essential, as in Figure 4. 3. All genuine nodes in the model are assumed to have the same processing power, storage space, communication speed, and activation energy [165]. Attempts by malicious nodes to steal the security key of the base station and clusters are speculated to be more successful than those made by the legal node. The malicious actor causes havoc by creating a copy of the legitimate node in the network.

Figure 4. 4 shows a model of a Sybil attack using three groups of wireless nodes. One form of a network attack is a Sybil attack, which occurs when a malicious node shows many forged or false identities to other sensor nodes [166]. This is done by either legitimately taking the identities of other sensor nodes or by autonomously inventing new identities. Sybil nodes can disrupt WSN processes like multipath routing, which employs multiple possible paths to determine the optimal one by creating an arbitrary number of bogus node identities.

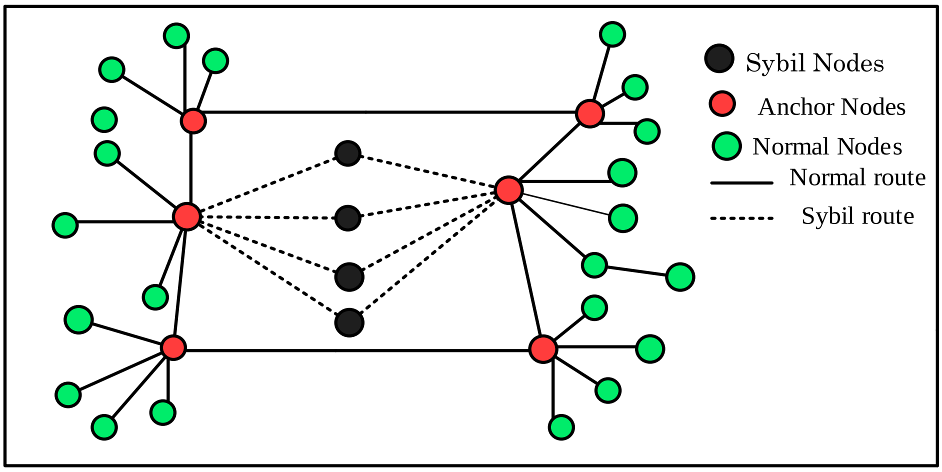


Figure 4. 4. Illustration of routing Sybil attack in wireless sensor networks [2].

In the attack model, the malicious nodes execute a series of simulated attacks against the genuine node's position and location using a variety of forged identities and routing paths. This reduces the network's lifetime by slowing down the authorized nodes' processing power.

4.5 Localization Techniques

Localization is essential in many uses of wireless sensor networks (WSNs) because it allows a target to be located by comparing the signal intensities of transmitters and receivers already in place [167], [168]. Certain methods are required for exact target localization and for locating and evaluating the nodes' locations and positions. Range-based and range-free localization techniques are separated out in the proposal. Financially, the second choice makes more sense but requires specialized gear. The received signal strength indicator (RSSI) and distance vector hop localization methods are analyzed for their use in localizing and situating wireless sensor nodes. It is necessary to apply the distance vector localization procedure to ascertain the relative positions of the sensor nodes and cluster leaders concerning the beacon nodes [10]. The approach computes and adjusts the distance to the unknown nodes. In WSN, gaps between beacon nodes can be located using the distance vector hop process. According to the distance vector method, the average hop size determines the shortest possible distance. The scheme was the first to discover this algorithm [3]. An example of a range-free localization method is distance vectors [62] with a series of steps.

Routing initialization: A message is sent from the beacon node to all of the sensors. The hop count associated with its position data is reset to zero when it enters the network [62]. Each node that receives such a message will add its own identification to the message, calculate its distance from the beacon node, and then send it on to its neighbours, increasing the hop count as it does so [169].

Calculating distance: We can calculate the distance to the unknown node and the average hop size with the aid of the beacon node. The unknown node receives a broadcast signal transmitted by the beacon node. After that, the unknown node sends a signal back to the beacon node, and the beacon node uses that information to pinpoint the unknown node's position. The distance between the beacon node and the unknown node can be determined by timing how long the signal travelled from the beacon node to the unknown node and back. If we know the distance the signal travelled from the beacon node to the mystery node, we can also determine the average hop size.

Position estimation: Finding the locations of unnamed nodes requires applying geometric calculations such as triangulation, the polygon approach, and trilateration [170]. The distance between two nodes can be determined in several ways, such as by using synchronization, radio signal intensity, or the physical characteristics of the carrying wave [171]. With the minimum hop count given by equation (4.1), we can determine the average hop distance from the anchor node to any other beacon.

$$HS_{ij}=\frac{\sum_{i\neq j}\sqrt{(u_i-u_j)^2+(v_i-v_j)^2}}{\sum_{i\neq j}hij} \tag{4.1}$$

Where i and j true anchor nodes, (u_i,v_i), (u_j,v_j), the known and true co-ordinates for i and j,

hij Hop counts of the anchor nodes, and HS_i average hop distance

The anchor node transmits its information, followed by hop-size calculation [3]. The distance between the sensor node and anchor computed with hop-size details is given by equation (4.2)

$$D_{pk}=H_pSi_{ij}\times hp_{pk} \tag{4.2}$$

The polygon technique enables the estimation of each anonymous node's position (P). P is the spot of the unidentified node denoted as (u, v) and di, the space among anchor and indefinite nodes. The position of the strange node p, assuming n beacon nodes involved, is estimated by equation (4.3) [3].

$$\begin{aligned}&(u-u_1)^2+(v-v_1)^2=D_1^2\\&(u-u_2)^2+(v-v_2)^2=D_2^2\\&.\\&(u-u_n)^2+(v-v_n)^2=D_n^2\end{aligned} \tag{4.3}$$

We can get a set (n-1) of expressions subtracting from the first equations to make the system linear, given as depicted in equation (4.4):

$$\begin{aligned}&u_1^2+v_1^2-u_n^2-v_n^2-2(u-v_n)u-2(v_1-v_n)v=D_1^2-D_n^2\\&u_2^2+v_2^2-u_n^2-v_n^2-2(u_2-v_n)u-2(v_2-v_n)v=D_2^2-D_n^2\\&.\\&u_{n-1}^2+v_{n-1}^2-u_n^2-v_n^2-2(u_{n-1}-v_n)u-2(v_{n-1}-v_n)v=D_{n-1}^2-D_n^2\end{aligned} \tag{4.4}$$

Rearranging the previous equations into the formula of $ui=BA^{-1}$, where A, ui, and B are expressed separately in equations (4.5)-(4.7):

$$A=\left\{\begin{matrix}2(u-u_n) & 2(v_1-v_n)\\2(u_2-u_n) & 2(v_2-v_n)\\. & \\2(u_{n-1}-u_n) & 2(v_{n-1}-v_n)\end{matrix}\right\} \tag{4.5}$$

$$B = \begin{Bmatrix} u_1^2 + v^2 - u_n^2 - v^2 + D_n^2 - D_1^2 \\ u_1^2 + v^2 - u_n^2 - v^2 + D_n^2 - D_1^2 \\ . \\ u_{n-1}^2 + {v_{n-1}}^2 - u_n^2 - {v_n}^2 + D_n^2 - D_{n-1}^2 \end{Bmatrix} \tag{4.6}$$

$$\mathrm{U_i} = \begin{pmatrix} u \\ v \end{pmatrix} \tag{4.7}$$

The location of the node is computed by solving the least square method stated in equation (4.8).

$$\mathrm{U=(A'A)^{-1}A'B} \tag{4.8}$$

When compared to their range-based counterparts, localization algorithms based on received signal strength indicators (RSSI) have seen significant adoption in the academic community [172]. Features like RSSI measurement and data transmission to higher stack layers are commonplace in modern wireless sensor nodes. For RSSI-based localization, neither time synchronization between nodes nor Ultrawideband (UWB) radios for more precise time of arrival calculations nor antenna arrays are required. It's a simple and cheap method for accomplishing node localization regarding software and hardware. However, when it comes to exact localization in large-scale wireless sensor networks, the DV-Hop method completely ignores the importance of detecting the distances between the one-hop neighbour nodes.

Rather than relying just on the DV-Hop method to zero in on WSN nodes, the hybrid strategy uses an additional two steps. These stages are included to enhance the accuracy of localization and the detection of malicious nodes. We start by using the RSSI data to estimate the distances between the anchor nodes and their one-hop surrounding sensor nodes, as opposed to relying on the average hop distance as the original DV-Hop algorithm did. Since the MAC sub-layer in most modern wireless sensor nodes calculates the RSSI value for every received packet and communicates that value to higher layers, using the RSSI value does not require any specific hardware or additional expenses.

In order to determine the location coordinates of an unfocalized sensor node, it is necessary to make an estimate of the distance between the sensor node and at least three anchors. The following equation (4.9) can then be used to derive an approximation of the distance between the sensor node and the anchor node:

$$d = 10^{\frac{RSSI_{d_0} - RSSI_d + X_\delta}{10n}} \tag{4.9}$$

$$RSSI_d = RSSI_{d_0} - 10n \log \frac{d}{d_0} + X_\delta$$

Where $RSSI_{d_0}$ is the power of the received signal at the sensor node located at reference distance d_0, which is typically 1 meter away from the anchor node. The $RSSI_d$ represents the amount of signal power that had been sent to the sensor node (which was an unlocalized node) by the sender. Whereas d is the estimated distance between the unlocalized sensor node and an anchor node, and n is the value of the Pathloss exponent, which can range anywhere from 2.2 to 6.5 depending on the environment in which the signal is being transmitted, is also represented. X_δ, is the shadowing factor, also known as the random variation in RSS, is denoted by the letter X_δ and is a Gaussian distributed random variable that is measured in dB and has a zero-mean and δ is the standard deviation.

Second, once a sensor node N is found, it becomes an anchor and is used to locate further sensor nodes. Including additional (repurposed) anchor nodes is one way to enhance the precision with which the remaining sensor nodes can be localized. In wireless networks, where the number of anchor nodes is typically smaller, this is very helpful. In order to arrive at the best possible answer to a problem, evolutionary computers employ a method called differential evolution (DE), which involves repeatedly trying to improve a candidate solution with regard to some quality parameter. A metaheuristic is a method that searches through a large number of solutions without making any assumptions about the situation. Unfortunately, metaheuristics such as DE cannot guarantee that you will always obtain the optimal solution.

Traditional optimization approaches, such as gradient descent and quasi-newton methods, are inapplicable to optimization problems involving multidimensional real-valued functions, but DE is still applicable because it does not rely on the gradient of the optimization problem. This is why optimization problems that are intrinsically non-continuous, noisy, dynamic, etc., are ideal candidates for DE's use. DE's fundamental equations allow it to optimize a problem by maintaining a population of candidate solutions, creating new candidates by merging existing ones, and finally retaining the candidate solution with the greatest score or fitness on the optimization task at hand. Hence, the gradient is not required as the optimization problem is considered a black box that only returns a quality measure once a candidate solution has been provided.

Secure localization of malicious nodes is an important problem in wireless sensor networks. Differential evolution (DE), Butterfly optimization algorithm (BOA), and modified Archimedes optimization algorithm (MAOA) are three popular optimization algorithms that have been used for this purpose. DE is a population-based optimization algorithm that maintains a population of candidate solutions and iteratively improves them through mutation and crossover. The behaviour of butterfly swarms inspires BOA and utilizes the principles of swarm intelligence to optimize solutions. MAOA is a deterministic optimization algorithm that uses a spiral motion to move toward the optimal solution. Here is a comparison of these three algorithms:

Differential evolution algorithm (DE): DE is a stochastic optimization algorithm that is known for its simplicity and efficiency. It has been widely used in wireless sensor networks for the secure localization of malicious nodes. DE works by iteratively improving a population of candidate solutions using a set of operators such as mutation, crossover, and selection. DE has been shown to be effective in achieving accurate localization results with a low computational cost.

Butterfly optimization algorithm (BOA): BOA is a new optimization algorithm inspired by butterflies' behavior. BOA works by simulating the behavior of butterflies in searching for food sources. BOA has been applied to several optimization problems, including wireless sensor network localization. BOA has been shown to achieve accurate localization results with a relatively low computational cost.

Modified Archimedes optimization algorithm (MAOA): MAOA is an optimization algorithm that is inspired by the Archimedes principle. MAOA works by iteratively improving a set of candidate solutions using a set of operators such as mutation and selection. MAOA has been applied to several optimization problems, including wireless sensor network localization. MAOA has achieved accurate localization results with a relatively low computational cost.

Overall, DE, BOA, and MAOA are all effective optimization algorithms for the secure localization of malicious nodes in wireless sensor networks. The choice of the algorithm may depend on the specific requirements of the problem, such as the size of the network, the level of accuracy required, and the computational resources available. It is recommended to compare the performance of these algorithms using specific metrics and data sets to determine which algorithm is best suited for a particular problem.

4.6 Machine Learning Techniques

Data mining operations can be carried out on datasets using any number of data mining techniques that have been created [140]. Hybrid machine learning methods, including Naive Bayes (NB), Artificial Neural Networks (ANN), Decision Tree (DT), Extreme Gradient Boosting (XGB), Extra Tree (ET), Random Forest (RF), Ensemble Stacking (ES), and cluster labelling K-Means (CLK-M), are used to detect and categorize attacks. Machine learning allows systems to automatically improve or learn from study or experience and take action without the need for explicit programming [119]. There was no doubt that ML was enhancing the reliability, efficiency, and cost-effectiveness of our computational procedures. Through the use of automated, quick, and exact processing of progressively more complex data, machine learning (ML) develops models. The extracted or chosen feature set is then used in conjunction with machine learning methods to develop the classification algorithm. Supervised learning methods are used if the data are acquired via the standard fingerprinting approach. The alternative is to use unsupervised or semi-supervised learning methods, which could be appropriate for crowdsourced data. When applied to test datasets, these classification models can closely approximate a user's or object's actual position. Varied data mining algorithms take extremely different techniques when operating on datasets. According on the structure of the datasets, the efficiency of these methods varies. Selecting the right method for the data's structure is crucial for optimal performance. Here, we provide a high-level overview of the data mining methods employed in classification procedures on a selection of representative benchmark datasets.

Machine learning (ML) is a class of artificial intelligence applied to handle various problems, including regression, classification, optimization, and clustering in engineering and computing applications. ML techniques are used in WSNs to reduce problem complexity and enhance overall network performance by extracting new features and information from the network traffic. Machine learning techniques are also used for dynamic routing handling for network performance enhancement. Machine learning processes and automatically improves the model without explicitly programming [173]. ML produces models by automatically analyzing accurately and quickly, even with more complex data.

Machine learning techniques are categorized into supervised, unsupervised, semi-supervised, and reinforcement learning techniques [121]. Supervised learning is the most data processing technique in machine learning. The sample of inputs and outputs of the datasets having labels are provided, and it finds the relationship among them by training model for the system. No outputs (unlabeled) are associated with the inputs; the model extracts the relationships from the data. This technique is used to classify patterns into clusters, dimensionality reduction, and anomaly detection from the data. Figure 4. 5 shows the various machine-learning techniques and their applications in wireless sensor networks.

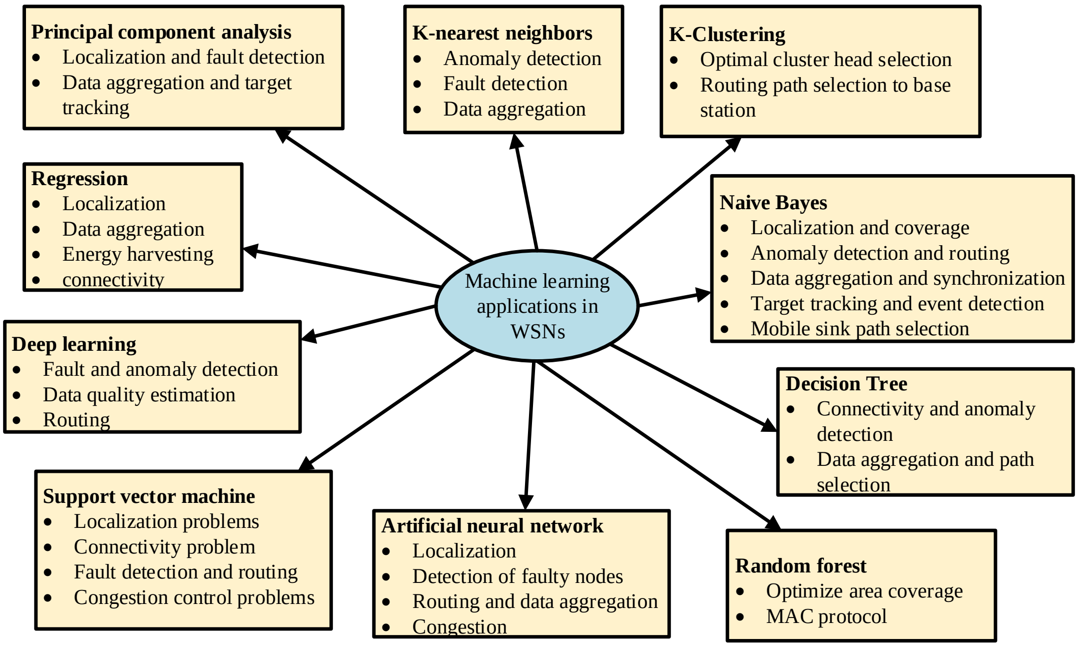

Machine learning applications in WSNs
Principal component analysis
• Localization and fault detection
• Data aggregation and target tracking
K-nearest neighbors
• Anomaly detection
• Fault detection
• Data aggregation
K-Clustering
• Optimal cluster head selection
• Routing path selection to base station
Regression
• Localization
• Data aggregation
• Energy harvesting
• connectivity
Naive Bayes
• Localization and coverage
• Anomaly detection and routing
• Data aggregation and synchronization
• Target tracking and event detection
• Mobile sink path selection
Deep learning
• Fault and anomaly detection
• Data quality estimation
• Routing
Decision Tree
• Connectivity and anomaly detection
• Data aggregation and path selection
Support vector machine
• Localization problems
• Connectivity problem
• Fault detection and routing
• Congestion control problems
Artificial neural network
• Localization
• Detection of faulty nodes
• Routing and data aggregation
• Congestion
Random forest
• Optimize area coverage
• MAC protocol

Figure 4. 5. Application of the machine learning methods used in WSNs applied in a number of different ways.

4.6.1 Artificial Neural Networks

The multilayer perceptron artificial neural network, often known as the MLPANN, is a method of supervised machine learning that employs a human neuron model for the purpose of data classification [173]. ANN can process and produce accurate information by utilizing a large number of neurons and a data-based neuron model that can classify the input and offer the proper output. ANN also has layers, linking nodes, and active duty in its structure [174]. Nodes having an activation function link the layers of an ANN. You can change the number of nodes and the number of hidden layers in a conventional ANN by adjusting the trainable parameters, as shown in Figure 4. 6. The efficiency of many different schemes can be increased with the help of ANN. This includes the identification and localization of sensor nodes, as well as routing and congestion control, and data aggregation in WSN. Data-driven Artificial Neural

Networks are useful for illustrating the dynamics of nonlinear systems. They perform well when identifying and modeling nonlinear systems. Conventional approximation abilities and adaptable structures allow them to represent nonlinear features [130] accurately. DoS assaults are represented by y1, y2,..., yn, with x1, x2,..., xn being the input parameters of the benchmark dataset containing multiple protocols, services, and the identification of the nodes. The ANN is used to identify malicious nodes' locations and pinpoint where sensor nodes are [61].

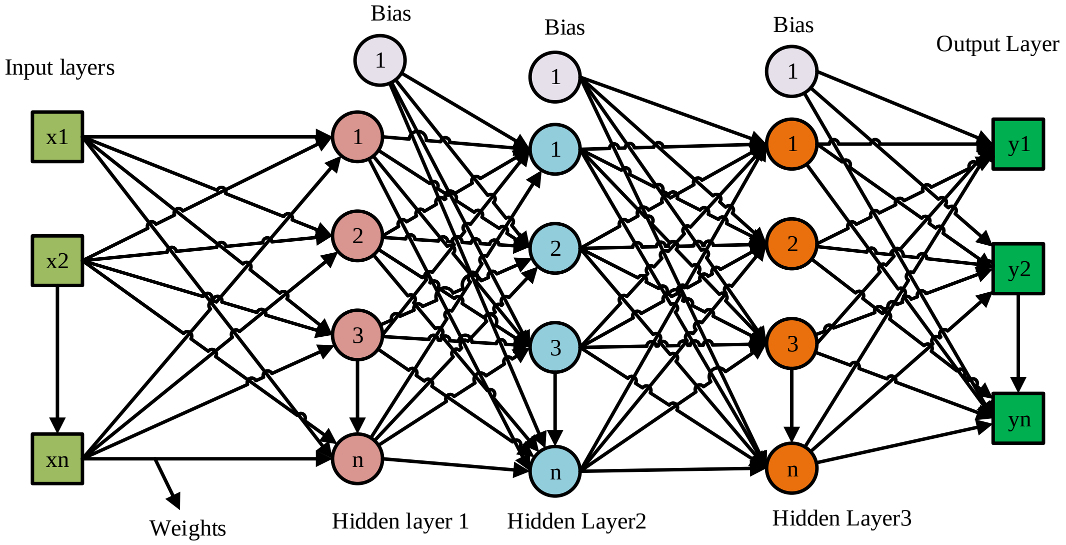


Figure 4. 6. The structure of the MLPANN Architecture for the proposed scheme includes three concealed levels [50].

Input nodes, hidden layers, bias and output nodes, and linking neurons are just some of the trainable characteristics that allow for precise attack localization and detection in the proposed approach. Using artificial neural networks (ANNs), WSNs gain superior computational intelligence for scalable and adaptive features [164]. The multilayer perceptron of artificial neural networks was used to locate the sensor node with pinpoint accuracy. Accurate node positions and locations in WSNs can be used for prediction and clustering.

4.6.2 ANN Applications in WSNs

Identifying network traffic is a growing area of interest in network administration, attracting scholars worldwide [175]. Its increase is directly correlated with the expansion of the network's capacity. Several of the older methods of traffic identification, including port-based or deep packet inspection, are ineffective when dealing with new types of network operations like peer-to-peer file sharing (P2P). Improved recognition efficiency and identification accuracy may be achieved by ML-based traffic identification by using a feature selection approach. That can pick the best features in response to the impact of the great traffic behaviour characteristics (Machine Learning). The Multilayer Perceptron artificial neural network (MLPANN) technique is superior to other precision-identifying methods. It has been demonstrated that increasing the number of training samples can achieve a higher identification rate. Yet, MLP has benefits and downsides of its own, which necessitate improvements.

Recent research efforts have focused on developing neural network–based fingerprint localization techniques (ANNs). An important benefit of using an ANN is that it can provide accurate node position recognition even when RSSI values are contaminated by noise. While utilizing ANNs, having precise information about the indoor environment or the locations of reference nodes is unnecessary. ANN interpolates the fingerprint database acquisitions to approximate a mapping between the fingerprint space's dimensionality and the node coordinates. During the ANN's training process, the collected RSSI vectors are used to fine-tune the weights of connections between neurons. While training may take some time, the speed at which the node can be localized is much greater than any analytical estimate of its position.

The multilayer perceptron is currently the most popular ANN architecture (MLP) for localization applications, including range-free wireless sensor nodes. WSN utilized the MLPANN to implement fingerprint-based localization. Specifically, we compared the accuracy of 43 different backpropagation training algorithms. A strategy quite similar to that was proposed. The ANN has regularly refreshed training to adapt to new wireless channel parameters. They presented an ensemble of four MLPANNs with varying numbers of inputs. An ANN with the same number of inputs as the currently linked reference nodes are chosen and issued if the localization operation must be carried out, as specified by this strategy. In light of this method's limited scalability, we decided to limit reference nodes to a maximum of four connections per. Localization results obtained by the ANNs ensemble were found to be more accurate than those provided by approaches based on fuzzy learning systems or genetic algorithms. We propose a cooperatively optimized and secure Multilayer perceptron artificial neural network (MLPANN) that combines range-based and range-free localization strategies for scalable and wide-area networks.

Figure 4. 7 illustrates how AI-based methods can be applied to common wireless sensor network (WSN) issues such as data aggregation and fusion, routing, task scheduling, optimal deployment, and localization [176]. In this context, computational intelligence is a branch of machine learning that uses various biologically inspired methods for making predictions, including neural networks, fuzzy systems, and evolutionary algorithms. This learning process could be constructed with cascading decision chains to recognise non-linear and complex functions.

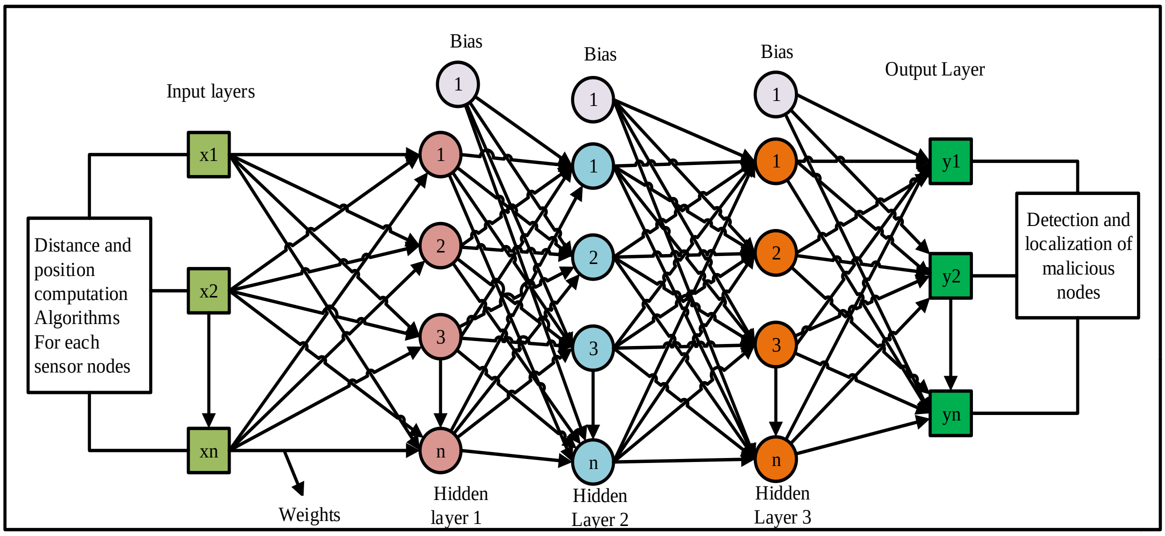


Figure 4. 7. Illustration of MLPANN-based localization algorithms for detecting and locating assaults on WSNs.

To yet, distributed neural networks have not seen widespread application in WSNs due to their high computing needs for learning the network weights and extensive administrative overhead. Yet, neural networks' capacity to learn many outputs and decision boundaries in centralized solutions makes them ideally suited for dealing with a wide variety of network problems with a single model.

By estimating the distance and position of each node type with unique identification, a multilayer perceptron artificial neural network is constructed to detect and localize malicious nodes, as demonstrated in Figure 4. 8. Both homogeneous and heterogeneous wireless sensor networks are envisaged for the sensor nodes in our aim. The beacon nodes have high computational data processing and localization to estimate and compute the location and position of the conventional sensor nodes in the network. Range-free node positioning gains accuracy when machine learning is used for WSN localization [177]. Particularly, range-free localization algorithms benefit greatly from incorporating artificial neural networks (ANNs), which vastly outperform their predecessors in accuracy and efficiency. To build a model that accurately generalizes to data not included in the training set, the MLPANN learning technique that begins with a labelled data set is required before any tweaks to the weights can be made [178].

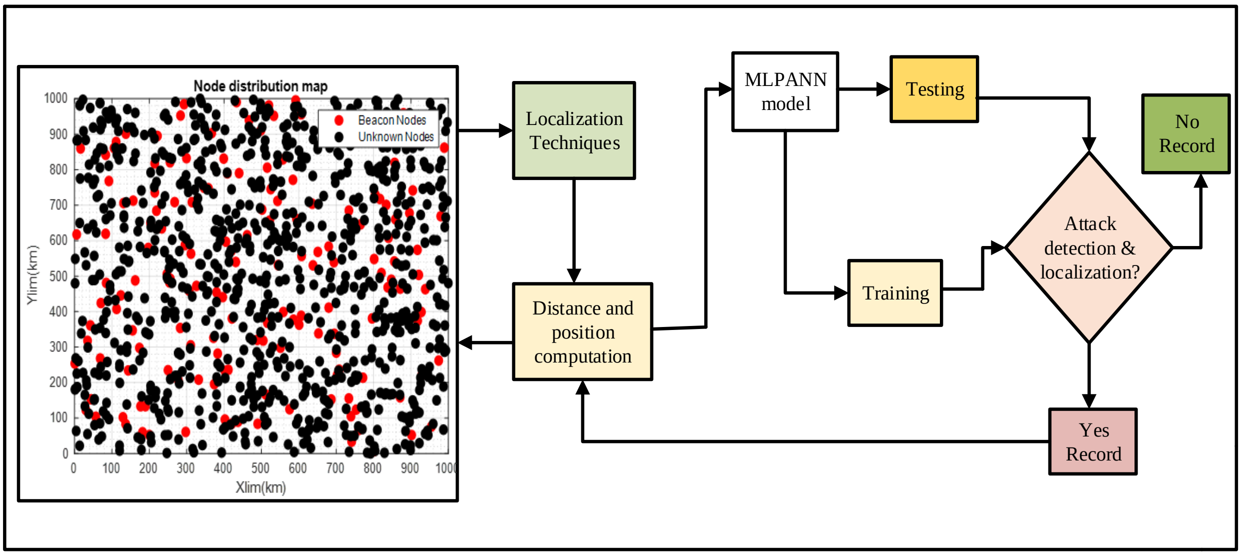


Figure 4. 8. Techniques of secure localization for identifying and localizing malicious attacks in wireless sensor networks (WSNs) employing MLPANN [36].

4.6.3 Benchmark datasets

Here we evaluate the efficiency of attack detection and localization accuracy using three benchmark datasets: UNSW-NB 15, WSN-DS, and NSL-KDD. Cyber LAB used the IXIA Perfect Storm tool for cyber security to develop attack behaviour in the raw network packets that comprise the UNSW-NB 15 dataset [179]. It is illustrated how the cyber security dataset is divided into training and testing samples and how the total error for each weight is updated using the batch mode [180] as in Table 4. 1. There are ten distinct types of attacks in the dataset, each with a unique frequency distribution.

Table 4. 1. Frequency distribution of DoS attacks for training and testing in the dataset [41].

Attack class	Frequency	Percent	Attack class	Frequency	Percent
Analysis	29	0.3	Generic	147	1.5
Backdoor	21	0.2	Shellcode	48	0.5
DoS	382	3.8	Reconnaissance	97	1.0
Normal	8632	86.0	Exploits	538	5.4
Fuzzers	122	1.2	Worms	17	0.2
Total	9186	91.5	Total	847	8.6

The proposed system's data collection contains various assault operations that must be processed and classified. Normal, Shellcode, Analysis, Backdoor(s), DoS, Exploits, Fuzzers, Reconnaissance, Generic, and Worms are the various types of assaults [179], [181]–[185].

The most common methods of routing attacks in the WSN-DS dataset are described in Table 4. 2 and are used to compare the effectiveness of the suggested to something else. Eighty percent of the 84556 data points and 23 features are used for training, while twenty percent are used for testing while developing a predictive model.

Table 4. 2. Frequency distribution of DoS Attack in WSN-DS and NSL-KDD dataset.

WSN-DS dataset			NSL-KDD dataset		
Attack type	Frequency	Percent	Attack Type	Frequency	Percent
Blackhole	2607	3.1	DoS	37403	33.7
Flooding	1019	1.2	normal	61355	55.2
Grayhole	3287	3.9	Probes	10600	9.5
Normal	75300	89.1	R2L	913	.8
Scheduling	2343	2.8	U2R	839	.8
Total	84556	100.0	Total	111110	100.0

The NSL-KDD is another benchmark dataset for assessing the suggested method; it has 100069 samples and includes kinds of attacks like denial-of-service (DoS), Probes, user-to-root (U2R), root-to-local (R2L), and normal. The results as in Table 4. 2. There are 41 features in the dataset: 38 numerical and three categories.

4.7 Proposed System

Design, planning, deployment, data processing, training, testing, attack classification, attack detection, and localization are the phases of the proposed system. The processing phase involves normalization and feature selection regarding network traffic security data. The proposed system shown in Figure 4. 9 is created with the help of an AI Multilayer Perceptron (MLPANN). Weights in an MLP are determined by backpropagation of error, making it a feed-forward artificial neural network [186]. The Artificial Neural Network (ANN) method is an IPU-based decision-making model that learns stochastically [187]. ANN can infer the nonlinear connection between inputs and outputs and chart the data flow between nodes. The multilayer perceptron (MLP) is shown in Figure 4. 9, set up with incoming data layers, three intermediate ones, and a final output layer. The suggested system leveraged the optimization techniques of gradient descent to speed up and improve the precision with which it could detect and localize attacks. The multilayer perceptron is used in both the training and testing phases, and the approach is statically driven.

Several mechanisms are built into the proposed architecture to detect malicious or unexpected routing. The first steps in the procedure involve collecting and preprocessing network data [188]. The next step is to locate any blanks before processing begins and then populate them with the correct data. The average is what we always fall back on. After that, duplicate values are removed from the dataset to make it more presentable. The next step is to encode the information and then normalize it. The dimension reduction process is used for the encoded data to make it easier to work with. Anomaly detection can be aided by performing feature optimization to draw out the most informative aspects of the data. Selecting the most relevant features from a dataset is essential for identifying anomalies. It helps reduce the computing cost of processing the same amount of data. To calculate entropy (E), use the equation given below as in (4.10):

$$E = -\sum_{i=1}^{L} P_i \log P_i \tag{4.10}$$

In which p is the likelihood that a given label belongs to a specific category after determining the best way to choose features for anomaly detection using a hybrid machine learning approach for intrusion detection in a wireless sensor networks.

Multilayer perceptron in an artificial neural network uses sigmoid and softmax activation algorithms. The output layer and the hidden layers are activation functions. Equations for the sigmoid and softmax functions are provided below (4.11) and (4.12), respectively.

$$y=\frac{1}{1+e^{-x}} \tag{4.11}$$

$$z=\frac{e^{x}}{\sum_{k=1}^{n} e^{x}} \tag{4.12}$$

Multilayer perceptron outputs a vector with k indexes and n elements, where y is the network's reaction to the input x. The softmax z is used as an activation function for the classifier's final processing layer. Our scenario evaluates performance using a structure with one to three hidden layers. Next, the sampling dataset will be reduced so that the feature to be extracted can be pinpointed with more precision [189]. Use pooling methods to accomplish this. By pooling, we may reduce the image size and the required computations. Max Pooling was used as the method of choice. Max Pooling uses the maximum value of the feature maps to create a new map. A node's activation function determines its output's value in response to a given input or set of inputs.

4.7.1 Optimization and Tuning Techniques

The optimization step of an ANN seeks to find the best weighting scheme possible. Considered a persistent nonlinear optimization problem, this is a challenging optimization topic. Algorithms are plentiful in print media. One of the most often used algorithms is backpropagation. Despite its success, this last option may encounter a local minimum problem. We have combined a regional search strategy with a differential evolution algorithm to solve this problem and increase the probability of fast convergence. Using Adam Optimizer, we can efficiently adjust the network's weights during training to reflect the model's evolving understanding of its parameters. According to the authors, two existing modifications of stochastic gradient descent, the Adaptive Gradient Algorithm (AdaGrad) and the Root Mean Squared Propagation Algorithm, have been merged into a single algorithm called Adam (RMSProp). These two methods are similar because they keep the same learning rate regardless of the values for any given parameter. AdaGrad and RMSProp are helpful tools, in Adam's opinion [189]. The loss function gradient and gradient descent optimization are computed to fine-tune neuron weights. Specifically, the networks are trained using the gradient descent non-linear optimization technique and a gradient-based algorithm [163]. Training times for an artificial neural network's multilayer perceptron can be reduced with the help of the gradient descent algorithm. The technique is also useful for bringing together the network's weight iterations.

Multilayer perceptron requires the use of optimal weights, the use of the ideal number of hidden layers and hidden nodes, and the use of the optimal collection of important attributes [190]. The output data are weighted by layer and gathered at a hidden node. A bias node is also assigned a weight value. A nonlinear activation function is applied to the sum of the weighted input values. The only stipulations are that the nonlinear function is differentiable and that its output values are contained in an interval. The objective of the MLP optimization problem is to determine the ideal set of weights such that the estimated and real outputs are as close as possible. To model this problem, we turn to continuous optimization. They are using the Optimization Problem Classification to guide the solution process.

4.7.2 LSTM-FFNN Technique

The proposed optimized multilayer perceptron artificial neural network used Long Short-Term Memory and Feed-Forward Neural Networks(LSTM-FFNN). The Long Short-Term Memory (LSTM) RNN operates similarly to a recursive function that calls itself within the context of DL. The term "recurrent" refers to the fact that a recurrent neural network applies the same computation to each data point in a recursive manner. There are gradient vanishing and explosion problems with the RNN. Unlike other DL methods like Deep NNs, the LSTM can understand causal relationships between events in a time series and retain relevant information from previous iterations for use in present and future predictions. We assume that the model's inputs are the three times steps before the current one. As seen in the unfolded version, the information from the first module feeds into the second [191], [192].

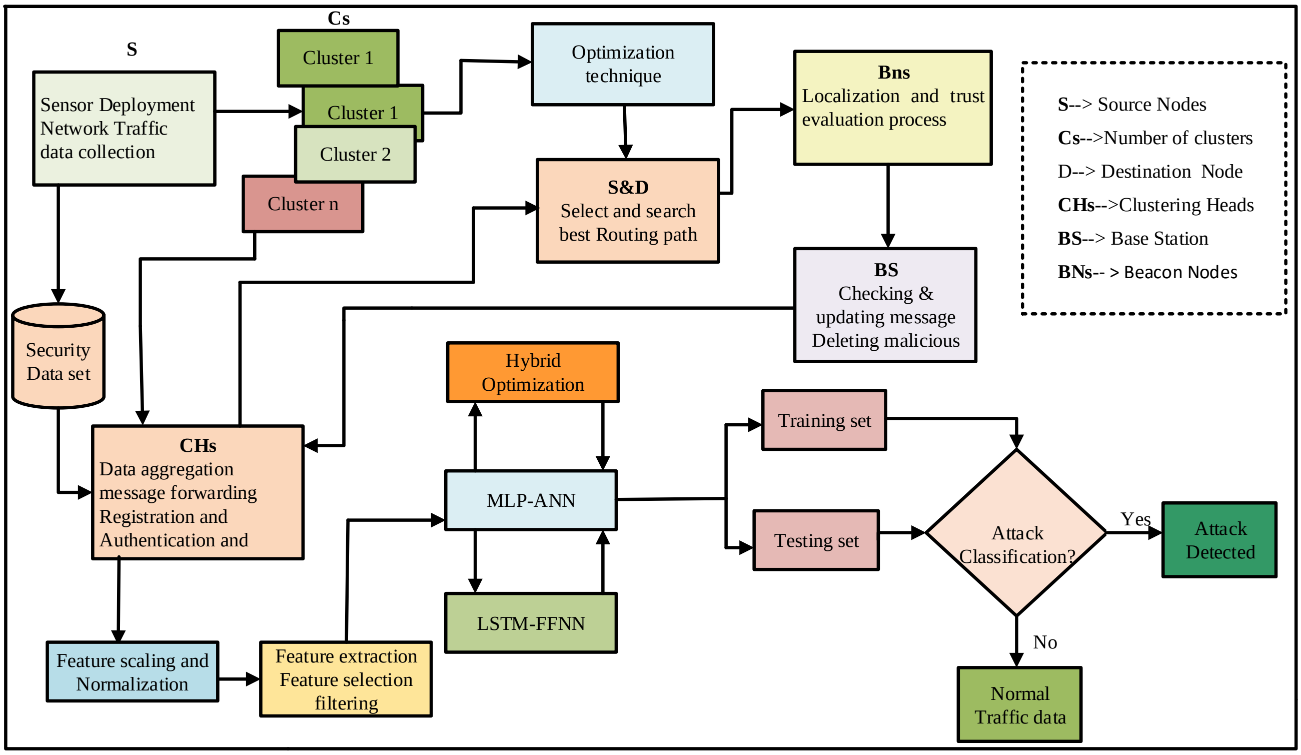


Figure 4. 9. A block diagram showing the proposed attack localization and detection strategy for wireless sensor networks (WSNs).

The Fast Forward Neural Network (FFNN) is an example of a NN that, unlike the Long Short-Term Memory-like LSTM, does not use historical data to inform its predictions. Everything their forecasts are based on is data from the current lag. The FFNN is comprised of inputs and a hidden layer with n-nodes. The output of a node is proportional to the inputs it receives and the strength of the links between it and other nodes. The five layers that make up our model are a vector input layer, three hidden levels, and a single-node output layer that delivers a 1 or 0 depending on the classification task at hand.

In this work, we experimented with several different activation functions while considering the threshold. In this case, we employ the ReLU activation function. It stands for "Rectified Linear Unit," which describes a component of non-linear processing. The purpose of providing non-linearity to the network is accomplished, as the real world is often quite non-linear. In a mathematical sense, it can be defined as in equation (4.13) (16) shown below:

$$\mathrm{ReLu}(x) = \begin{cases} x & \text{if } \mathrm{x} \geq 0, \\ 0 & \text{if } \mathrm{x}<0 \end{cases} \tag{4.13}$$

The z value was utilized as input to a logistic function, producing numbers between 0 and 1 with a 0.5 threshold using the Sigmoid activation function (or logistic function). Mathematically, this may be described as (4.14):

$$\delta(\mathrm{z}) = \frac{1}{1+e^{\text{-x}}} \tag{4.14}$$

Two types of labels (outputs) guide the selection of an activation function for this approach. For this reason, the binary classification approach is the best option. We used cross-entropy, which is a standard loss function for ANNs. Specifically, [50] defines cross-entropy (C) as in (4.15):

$$\mathrm{C} = -\frac{1}{\mathrm{n}} \sum_{\mathrm{x}}^{\mathrm{n}} \left[\mathrm{ylna} + (1-y)\ln(1-\mathrm{a}) \right] \tag{4.15}$$

$$\mathrm{a} = \delta(\mathrm{z}) = \delta\left(\sum_{\mathrm{j}} \mathrm{w}_{\mathrm{j}} \mathrm{x}_{\mathrm{j}} + \mathrm{b} \right) \tag{4.16}$$

Where hybrid optimizer Adam will make small, incremental changes to the weights w and biases b. The Adam algorithm is a refinement of the Stochastic Gradient Descent method (SGD). The Scikit-learn documentation claims Adam does reasonably well on large datasets. In Adam, you can tweak four separate values: your learning rate, the decay rate of your first-moment estimates, the decay rate of your second-moment estimates, and a tiny amount to prevent a division by zero. Because it incorporates the best features of two other popular optimizers, Adam performs better than SGD in noisy settings (the adaptive gradient algorithm and root mean square propagation). To kick things off, we develop an initial framework for optimizing hyperparameters that prioritizes efficiency above raw performance.

4.7.3 Performance Evaluation Metrics

The effectiveness of the proposed system is assessed using the metrics of the confusion matrix. These include accuracy, sensitivity, specificity, and training time. The new proposed approach is evaluated the evaluation metrics, including power consumption, detection rate, network lifetime, and detection accuracy of the attacks in the network. We also measure the efficacy of the suggested system concerning the following metrics: network scalability; events and communication overhead; communication range; communication failure; communication failure rate; aggregation ratio; and network load.

4.7.3.1 Network lifetime

It is the operational time in which the network performs the dedicated task. It can be computed when the source node energy drains to transfer to the base station. This shows that the loss of nodes leads to the loss of network functionality [93]. Network lifetime in WSNs depends on many factors, some of which are security related while others are not. Security measures such as secure localization, routing protocols, and optimization techniques can help to increase the lifetime of WSNs, by reducing the amount of data that needs to be transmitted, as well as protecting the network from malicious attacks. Secure clustering and data aggregation techniques can also help reduce the amount of data that needs to be transmitted, thus increasing the network's lifetime. Finally, machine learning techniques can be used to identify and detect potential security threats and to adapt the network parameters to increase the overall security of the network, thus increasing the network's lifetime.

4.7.3.2 Accuracy and F-measure

Accuracy and F-measure are two of the most commonly used metrics for evaluating the performance of attack classification in IoT-based WSNs. Accuracy measures how accurately the system can identify and classify attacks. It is calculated by dividing the number of correctly identified attacks by the total number of attacks. F-measure measures how well the system can distinguish between different types of attacks. It is calculated by taking the harmonic mean of precision and recall, where precision is the ratio of true positives to total predicted positives, and recall is the ratio of true positives to total actual positives. Higher accuracy and F-measure scores indicate better performance in attack classification.

The reliability of a data transmission over time is measured in terms of the percentage of lost packets. How well a learning model works depends on how accurate its predictions are [193].

We conducted experiments to verify the efficacy of the suggested technique. We compared the results to those of a previously developed, secure blockchain that relied on a network of federated hybrid machine-learning models. This can be done by calculating the fraction of attacks that were correctly labelled as True Positive, the fraction of trusted nodes that were labelled as True Negative, the fraction of false positives that were incorrectly marked as True Negative, and the fraction of false negatives that were incorrectly labelled as True Positive.

4.7.3.3 Detection Rate

The detection rate is a positive proportion of correctly categorized normal traffic relative to the total number of samples in the collection. The true positive rate is calculated in equation (4.17). Accuracy is measured by the percentage of attacks correctly labelled compared to the total number of attack occurrences in the network. For each attack, the true positive (TP) count indicates how many times they were accurately identified as threats [194], [195], and a network's false positive (FP) rate is the percentage of attacks that were misidentified [196]. A network's "true negative" (TN) consists of all legitimate nodes properly classified as such, while a "false negative" (FN) mistakenly labels all honest nodes as malicious ones.

The Detection Accuracy is the fraction of trusted sensor nodes that correctly identify malicious nodes relative to the total number of harmful nodes in the network [6]. When compared to the A confusion matrix creates a baseline for calculating the parameter metrics. The number of instances is tabulated as four values: true negatives (TN), true positives (TP), false positives (FP), and false negatives (FN), and expressed as in equation (4.17).

$$\text{Detection Rate(DR)} = \frac{\text{TP}}{\text{TP+FN}} \tag{4.17}$$

$$\text{Recall (RC)} = \frac{\text{TP}}{\text{TP+FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{FP+FN}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{FN+TP}}$$

$$\text{False positive rate} = \frac{\text{FP}}{\text{FP+TN}}$$

The detection accuracy and F-Measure are also expressed as mathematically as shown below using equation (4.18):

$$\text{Accuracy(Acc.)} = \frac{\text{TN+TP}}{\text{TN+TP+FN+FP}} \tag{4.18}$$

$$\text{F-Measure} = \frac{2\times\text{R}\times\text{P}}{\left(\text{R+P}\right)}$$

To determine the False Negative Rate (FNR), which is the number of malicious nodes that the proposed model wrongly identifies as legitimate nodes, we use the following formula (4.19):

$$\mathrm{FNR}=\frac{\mathrm{FN}}{\mathrm{FN+TP}} \qquad (4.19)$$

In addition, this work presents new measures for evaluation, including the average localization error, coverage, localization, and detection accuracy.

4.7.3.4 Recall and Precision

Recall and precision are important metrics for evaluating how well an attack classification system works in IoT-based WSNs. Recall, also known as the true positive rate, measures how many of the attacks in the network were correctly identified by the system. Precision, also known as the positive predictive value, measures how many attacks the system identified were present in the network. A high recall rate indicates that the system can correctly classify most of the attacks in the network. In contrast, a high precision rate indicates the system can identify most attacks it detects correctly. Thus, a high recall and precision rate is necessary for an efficient attack classification system in IoT-based WSNs.

4.7.3.5 Localization Error

The localization error quantifies the distance between the estimated and actual locations [197]. The radio range of sensor nodes normalizes errors in localization to provide a consistent metric for comparison. Average localization error (ALE), Considering all of the nodes in the study area, we can calculate an average localization accuracy to characterize how well we can place them [198]. Several metrics measure the proposed system's efficiency, including the average detection rate, accuracy, precision, and recall. The ALE can be computed by adding up the LE of each unknown node and then dividing that sum by the total number of unknown nodes [170]. The LE measures how far away a node's predicted position is from its actual physical location. The average localization error (ALE) and average localization accuracy (ALA) are used as evaluation metrics. The average error localization, shortened as ALE [3], is computed in equation (4.20), respectively. The ALE is calculated by adding the LE of each unknown node and dividing it by the total number of unknown nodes. Specifically, the LE is defined as the discrepancy between the predicted and observed locations of black boxes.

$$\text{Localization Error(LE)} = \sqrt{\left(u_i' - u_i\right)^2 + \left(v_i' - v_i\right)^2} \tag{4.20}$$

$$\text{Average Localization Error(ALE)} = \sum_{i=1}^{n} \frac{\sqrt{\left(u_i' - u_i\right)^2 + \left(v_i' - v_i\right)^2}}{nR}$$

$$\text{Average Localization Accuracy(ALA)} = \left(1 - \left(\sum_{i=1}^{n} \frac{\sqrt{\left(u_i' - u_i\right)^2 + \left(v_i' - v_i\right)^2}}{nR}\right)\right) \times 100\%$$

Where $\left(u_i', v_i'\right)$ are the real coordinates of the anonymous node i and $\left(u_i, v_i\right)$ are the computed coordinates, and n denotes unknown nodes, and R is radius of communication in the network. The systems' performance is also evaluated by parameters including Speed, time to build the model (TTB), Relative mean square error (RMSE), Relative root square error (RRSE), Relative absolute error (RAE), Mean absolute error (MAE), Correctly classified (CC) and incorrectly detected (ICC) and Detection accuracy. The predicted values on the test samples are assumed to be $x_1, x_2, ..., x_n$ the actual values $y_1, y_2, ..., y_n$ [199]. The X_i numeric test samples for prediction. The MA error, as shown in the equation, is the mean of the individual errors ignoring the sign it is computed as in (4.21):

$$\text{MA Error} = \frac{\left|X_1 - \bar{Y}_1\right| + ... + \left|X_n - \bar{Y}_n\right|}{n} \tag{4.21}$$

Where X denotes the predicted values, Y is the actual values, $\bar{Y}$, is the absolute mean value, and n is the number of variables. The root means the square error is computed by the equation (4.22) for reducing the dimensionality of the predicted quantity given below:

$$\text{RMSE} = \sqrt{\frac{\left(X_1 - Y_1\right)^2 + ... + \left(X_n - Y_n\right)^2}{n}} \tag{4.22}$$

It is impossible to demonstrate the misunderstanding of the true values without the relative inaccuracies. Using the absolute error provided by the equation, we may normalize the errors projected values (4.23):

$$\text{Relative absolute error} = \frac{\left|X_1 - Y_1\right| + ... + \left|X_n - Y_n\right|}{\left|Y_1 - \bar{Y}\right| + ... + \left|Y_n - \bar{Y}\right|} \tag{4.23}$$

The root relative squared error (RRSE) for the data points in the dataset samples is computed by equation (4.24) as shown below:

$$\text{RRSE} = \sqrt{\frac{\left(X_1 - Y_1\right)^2 + ... + \left(X_n - Y_n\right)^n}{\left(X_1 - \bar{Y}\right)^2 + ... + \left(X_n - \bar{Y}\right)^2}} \tag{4.24}$$

4.8 Simulation and Testing

The primary experimental programming languages we used were Python 3 and MATLAB [79]. To evaluate the efficiency of the proposed technique on the dataset, we apply IBM SPSS and the WEKA Java toolboxes for data processing and analysis. The proposed method has been implemented using well-known libraries like NumPy, simplifying the manipulation of multi-dimensional arrays and matrices. Attacks can be analyzed and categorized with the help of matrices and arrays like these. Pandas provide easy access to potent analysis tools and the ability to manipulate data structures readily. Since their introduction, frameworks like TensorFlow and Keras have spread throughout the deep learning and machine learning communities. The Scikit-learn library can be used to create ML methods in both supervised and unsupervised settings. SMOTE was developed to select a larger sample size from under-represented demographics methodically. The models' networks were planned and simulated using MATLAB R2021a on a computer with an Intel(R) Xeon(R) Silver 4214 CPU at 2.20GHz 2.19 GHz (2 processors), with 128 GB of installed RAM (128 GB of usable RAM), running Windows 10 64-bit on an x64-based processor. Python is used for scripting, web development, and processing and analyzing data.

4.8.1 Localization Techniques

The primary focus of our research is to evaluate the efficacy of various hybrid-based enhancements to the original DV-Hop algorithm in identifying and isolating malicious nodes that have taken control of the beacon node and are providing erroneous routing information [200]. Our proposed algorithms have been fully implemented in the MATLAB simulator, where we have extensively tested and analyzed them for localization flaws and precision, as in Table 4. 3. MATLAB is a popular simulation programming language and numerical computing environment used by a wide range of scientists to conduct experiments, gather data, and develop models. Via four distinct topologies, we have analyzed how varying the proportion of anchor nodes, the total number of sensor nodes, and the nodes' communication range affect localization accuracy and localization error per node. An average localization error rate can be a proxy for an algorithm's accuracy in this area [169]. The cluster head selection process employs clustering and routing protocols to maximize the network's lifetime and enhance its performance. The simulated scenario utilizes routing assaults like sinkhole attacks, blackhole attacks, and Sybil attacks to gauge the precision of the localization and detection.

Table 4. 3. Simulation setup for the proposed network model based on range free and range based.

Parameter	values	Parameter	values
Number of sensors	300-1000	Number of clusters	10
Beacon nodes	60-120	Sink position	500, 1000
Unknown nodes	240-840	Number of attacks	5-60
Protocol type	Clustering and routing	Data size	4000 kb
Deployment Area	1000 ×1000 m^2	Attacks	Routing
Mobility	Random	Transmission radius	400 m

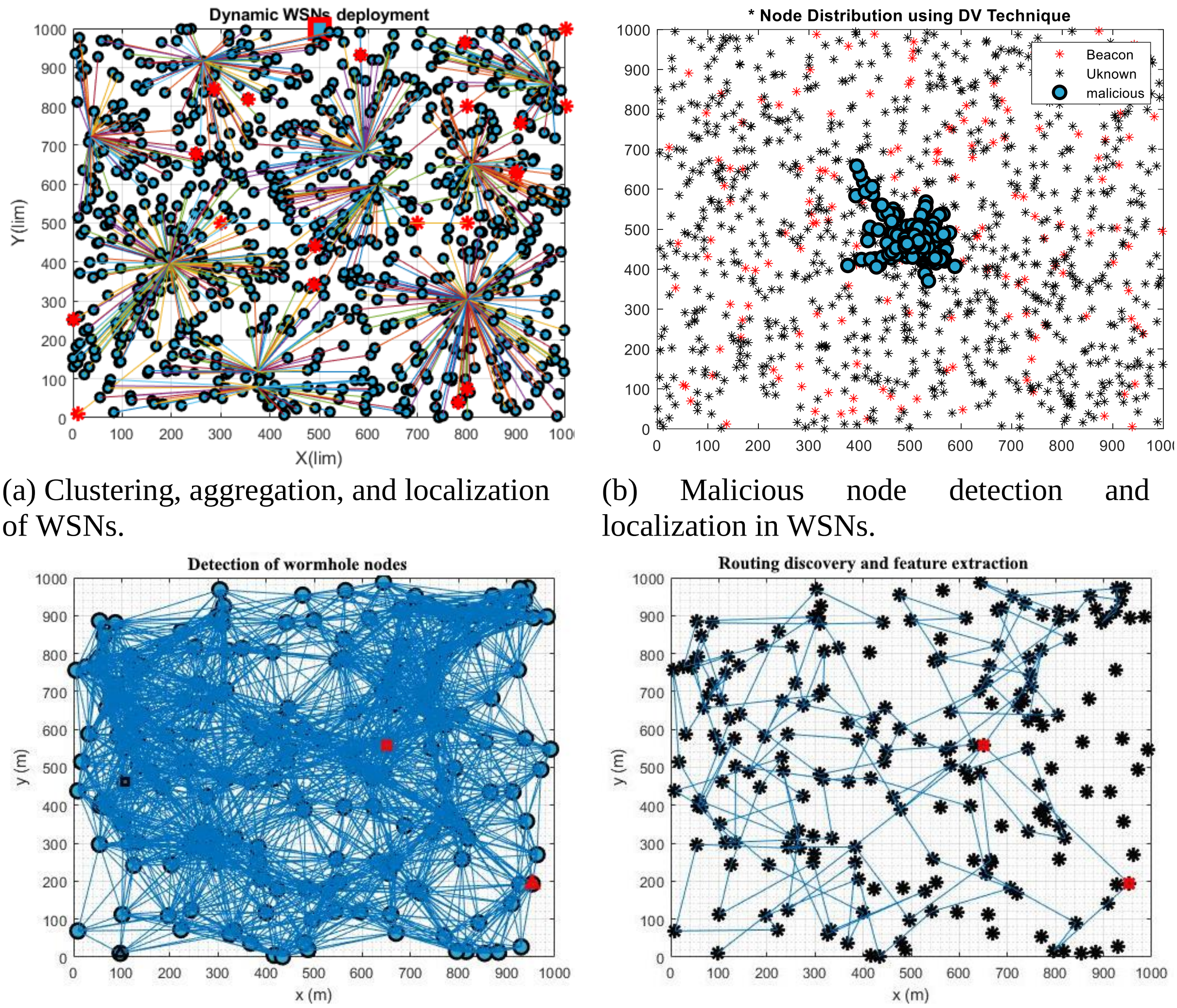
These parameters define the characteristics and configuration of the simulated network model. The range-free setup focuses on the number of sensors, beacon nodes, unknown nodes, deployment area, and mobility pattern. The range-based setup emphasizes the number of clusters, sink position, number of attacks, data size, type of attacks, and transmission radius.

4.8.2 Result and Discussion

In this section, the experimental results based on localization techniques and blockchain techniques are discussed in detail. The results of secure localization in WSNs are positive. Implementing secure localization ensures that the node’s location is accurately determined using the DV-hop algorithm, RSS, DE, and their combinations for secure communications and detection of malicious nodes. This allows for a high level of security against various types of attacks. Furthermore, secure localization allows the network to track the nodes' movements accurately. Secure localization allows the network to track the nodes and their activities accurately. Furthermore, secure localization can prevent malicious nodes from entering the network and stealing data.

4.8.2.1 Discussion on Localization Techniques

To simulate the deployment and localization of unknown wireless sensor nodes, a localization method involving beacon and sink nodes is used as in Figure 4. 10 (a). Figure 4. 10 (b) shows errors for unidentified sensor nodes. A localization approach is used to determine the position and error of each node. Accurately identifying and pinpointing the location of malicious nodes, with the aid of beacon nodes and the base station, is made possible by computing the localization accuracy for each node.



(a) Clustering, aggregation, and localization of WSNs.

(b) Malicious node detection and localization in WSNs.

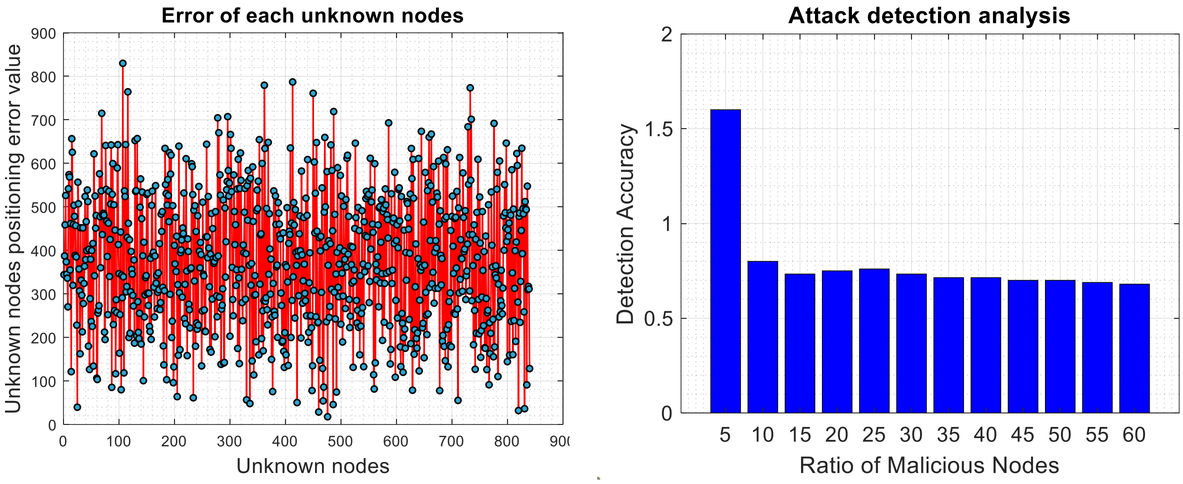
(c) Updating connection and detection of wormhole nodes.

(d) Routing discovery, feature extraction, and tunnel detection.

Figure 4. 10. Sensor node deployment using distance vector protocol and triangulation process for computing unknown node positioning in IoT-based WSNs.

Our model includes the following practical steps: model deployment and computation for unknown node localization error; routing discovery and feature extraction; registration, authentication, routing calculation; detection and classification of malicious nodes using hybrid federated machine learning models; and finally, malicious node detection and classification. Computing the distances and positions of the sensor nodes in red indicates wormhole assaults Figure 4. 10 (c). They create wormhole tunnels between two malicious nodes using route discovery and feature extraction, as in Figure 4. 10 (d). The simulation result demonstrates that after training on the extracted features, the suggested system can pinpoint the location of routing attacks with a 99.46% degree of accuracy, as in Figure 4. 10 (d).This demonstrates the superior localization precision of the wormhole assault when compared to T. H. Kim et a. [16], with a localization accuracy of 82.17%.

The efficiency, accuracy, and precision of the distance vector hop algorithm's localization are the yardsticks by which its performance is evaluated [201]. According to the results presented, the number of anchor nodes for its evaluation metrics is the determining factor in the practical localization estimation of the unknown and malicious nodes, as in Figure 4. 11. A node's relative error is the difference between its computed and actual positions. Malicious nodes impact the distribution and localization accuracy of nodes in WSNs because they generate the incorrect position and location of the unknown sensor nodes. When malicious nodes manipulate the routing paths and data of sensor nodes, it degrades the quality of service and performance of the entire network.



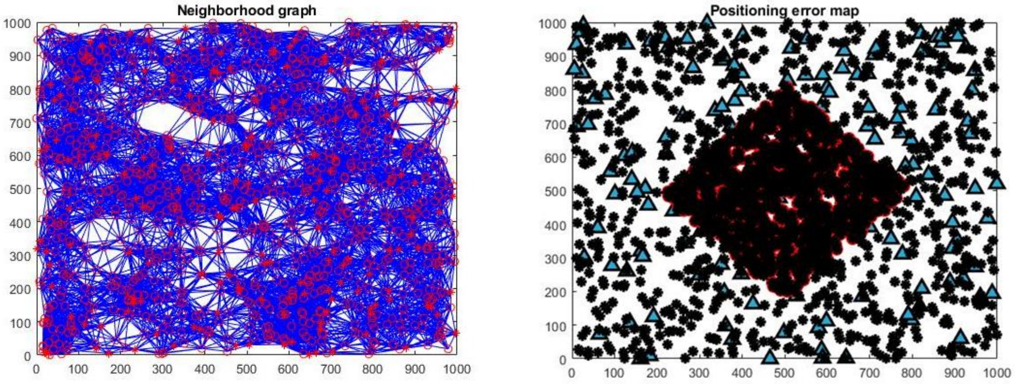
(a) Positioning error of unknown nodes

(b) Detection of the accuracy of malicious nodes

Figure 4. 11. Localization process for computing the faults in the localization and position of Unknown nodes and the detection accuracy of malicious nodes.

To accurately calculate hostile nodes, 840 unknown nodes, and 160 beacon nodes were used. The proposed system achieves an average localization accuracy of 99.51% and a detection accuracy of 99.83%, as demonstrated in Figure 4. 11 (a) and (b).

According to the simulation's findings, anchor nodes have more neighbors and a higher degree of connectedness than regular sensor nodes. If we use the regular model, we can determine that the average connectivity of the network is 404. The average number of neighbor nodes that each anchor node has is 63, as in Figure 4. 12 (a) and (b) The overall network's average localization error was reduced to 0.0049 thanks to the simulation's efforts, and this was achieved across all nodes.



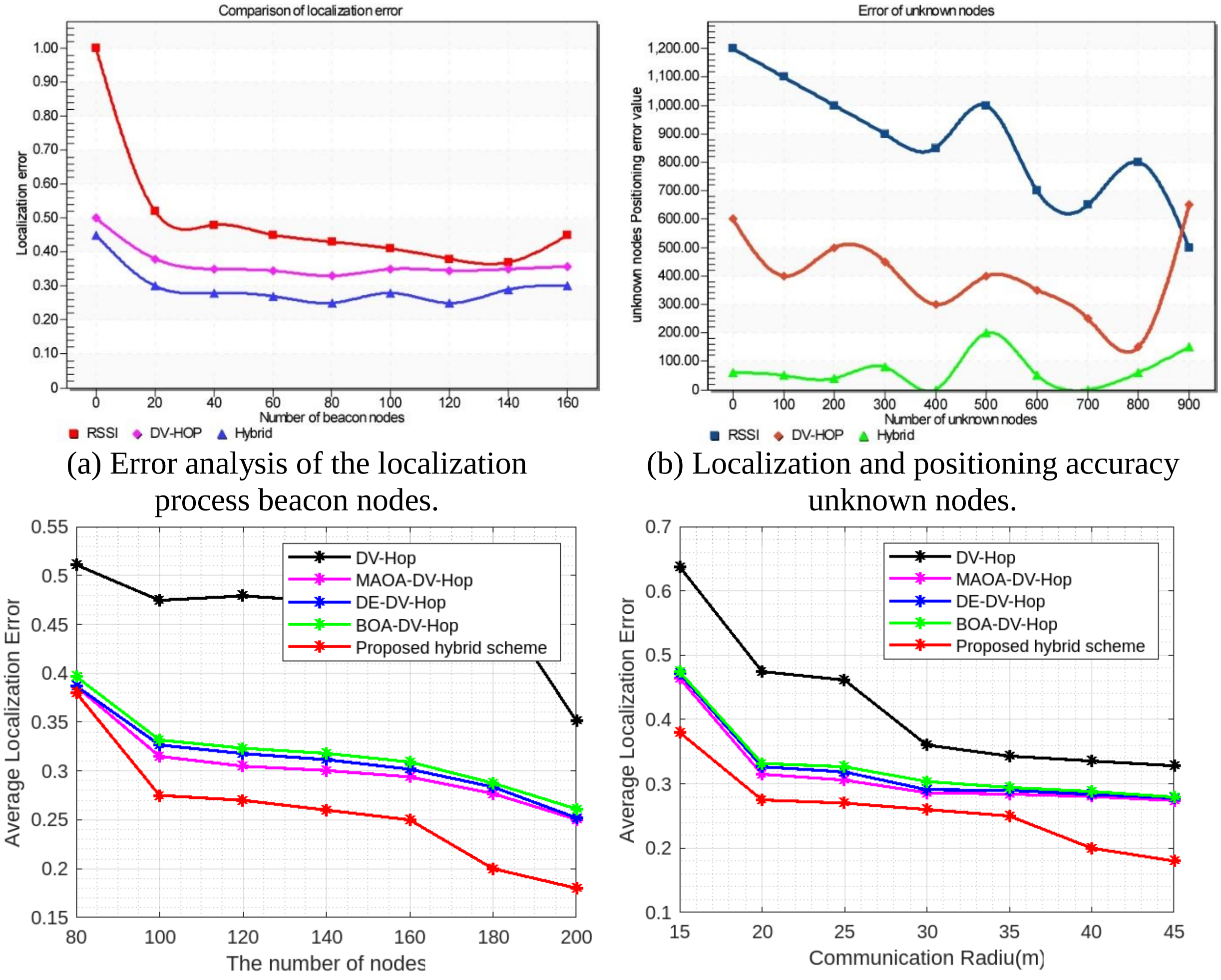
(a) Neighbor connectivity (b) Error positioning map

Figure 4. 12. Simulated sensor deployment and neighbour relationship diagram results based on localization techniques.

This would imply that all sensor nodes are precisely located and have a unique identity thanks to the beacon nodes, which help identify and localize malicious nodes.

Simulation results show that the proposed strategy uses a combination of range-free and range-based localization techniques to precisely establish the position and location of each unknown node with low power consumption. The suggested solution utilizes a network of twenty movable anchor nodes, which helps to reduce costs while also significantly improving the accuracy with which malicious nodes in WSNs may be located, as in Figure 4. 13 (a) and (b) to show how the suggested scheme's average localization accuracy is enhanced by a hybrid method that combines the DV-hop technique with different approaches like RSSI and DE for the beacon nodes and the unknown nodes, respectively.

To evaluate how well the proposed localization method works, we compared the localization error for several different hybrid approaches, such as the differential evolution distance vector hop (DE-DV-Hop) [202], the butterfly optimization algorithm (BOA-DV-Hop), the modified Archimedes optimization algorithm with distance vector hop (MAOA-DV-Hop), and the combination of received signal strength indicator with DE-DV-Hop (RSSI-DV-Hop) as shown in Figure 4. 13 (c) and (d). After only a few rounds, the RSSI-DE-DV-hop localization error curve is nearly horizontal across all five cases, indicating a high level of global optimization capabilities. Upon reaching the optimal local value, DE-DV-Hop quickly stabilized at 25 iterations. Therefore, RSSI-DE-DV-hop has superior localization performance compared to the other three localization methods, including MAOA-DV-hop. The results demonstrate that the differential evolution method helps to conclude that the hybrid range-based and range-free localization schemes improve localization accuracy and convergence rate.



(a) Error analysis of the localization process beacon nodes.

(b) Localization and positioning accuracy unknown nodes.

(c) The ALE concerning the node density

(d) ALE concerning communication radius

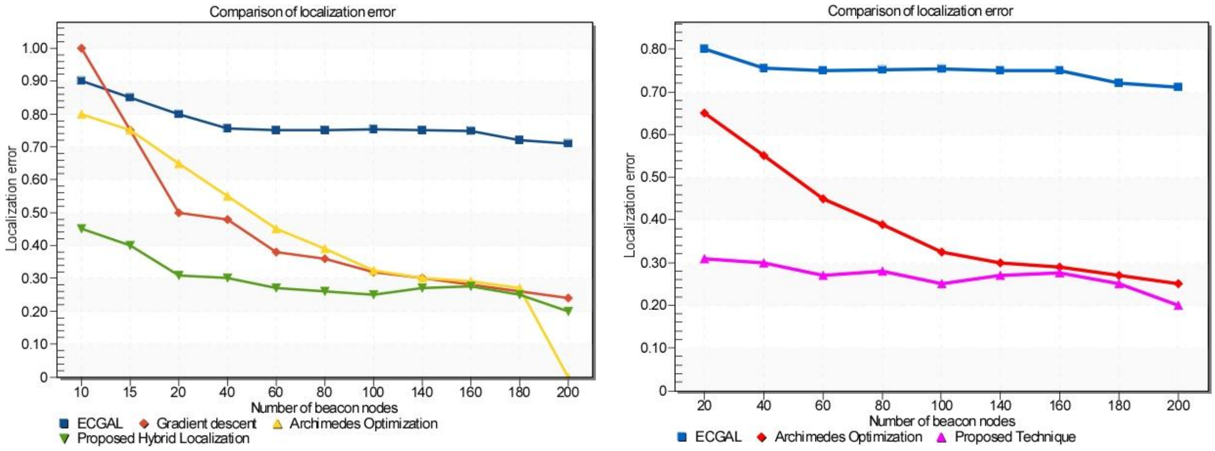
Figure 4. 13. Improving wireless sensor nodes' localization and positional accuracy using a hybrid method.

The hybrid DE-RSSI-DV-hop localization algorithm combines three different techniques, namely Differential Evolution (DE), Received Signal Strength Indicator (RSSI), and Distance Vector (DV)-hop, to achieve more accurate and secure localization in Wireless Sensor Networks (WSNs). The superior performance of this algorithm compared to DV-hop, DE-DV-hop, BOA-DV-hop, and MAOA-DV-hop can be attributed to the following reasons:

- DE algorithm: The DE algorithm is used to optimize the location estimation of the malicious nodes. This technique enables the algorithm to converge faster and provides more accurate estimates of the malicious node locations.
- RSSI technique: The RSSI technique uses the received signal strength of the nodes to calculate the distance between the nodes. This technique provides more accurate distance estimates, especially in environments with high interference or noise.

- DV-hop technique: The DV-hop technique estimates the distance between two nodes that are not directly connected. This technique is more secure than other techniques because it relies on multiple hops, which makes it difficult for an attacker to intercept or manipulate the messages.
- Hybrid approach: By combining these three techniques, the hybrid DE-RSSI-DV-hop algorithm can achieve better accuracy and security in localization. This hybrid approach ensures the algorithm is more robust and less vulnerable to attacks, making it more suitable for use in WSNs with malicious nodes. The hybrid DE-RSSI-DV-hop algorithm has better results than other localization algorithms because it combines multiple techniques to achieve better accuracy and security in WSNs with malicious nodes.

The results of the experiments measuring the location error about the number of beacons are shown in Figure 4. 14 (a) and (b). In addition, when the number of operational sensor nodes increases, the localization error lowers for all algorithms [168]. Regarding localization error, the proposed hybrid approach achieves the best results of any strategies we've tried. More reference points are found when there are 200 beacon nodes, bringing the localization error closer to zero.



(a) Comparison of localization schemes (b) Comparison with gradient descent

Figure 4. 14. Evaluation of the suggested strategy by comparing its localization error to existing methods by changing the number of nodes in the network.

Figure 4. 14 provides undeniable evidence that the novel approach is superior to standard location-based algorithms in identifying the cause of an error. Almost all strategies tested and evaluated have produced positive results when applied to the same setting. The proposed strategy permits a steady decrease since it provides more reference points for the target nodes.

Anchor nodes, on the other hand, fortify the network by bringing the unknown nodes closer to the anchors.

As may be shown in Figure 4. 14 (a) and (b), As more beacon nodes are added, the Average Localization Error (ALE) of four distinct localization methods goes down. We can more accurately estimate the typical hop length with more anchor nodes. Anchor nodes provide more reliable distance estimates when further nodes are added [202], [203]. This demonstrates that as the number of anchors increases, the suggested method becomes more accurate in estimating the locations of unknown nodes. Since some nodes can be used as anchors for node localization, the proposed method is more accurate than earlier approaches.

4.8.2.2 Discussion Based on Machine Learning

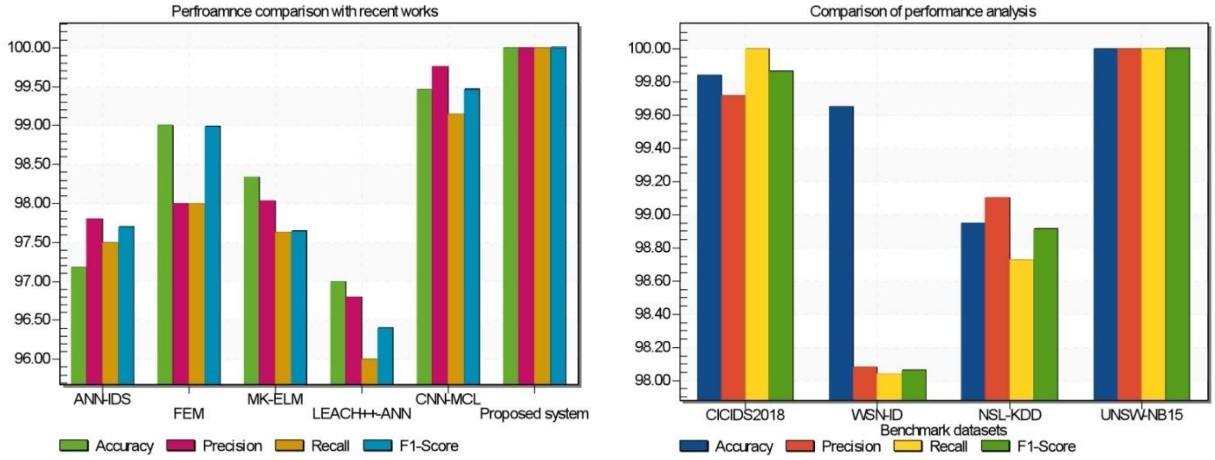
After extensive testing and comparison with other research utilizing multiple benchmark datasets, the performance of the proposed methodology is proven and validated. A comparison of the results of recent works, as in Table 4. 4, demonstrates that the improved MLP-ANN method is useful for spotting and localizing assaults in WSNs.

Table 4. 4. Comparative analysis of the proposed technique with recent attack detection models using the CICIDS2018 dataset.

References	Methods	Accuracy	Precision	Recall	F1-Score
[175]	MLP	97.56	99.12	98.79	98.23
[63]	ANN-IDS	97.18	97.8	97.5	97.69
[67]	FEM	99	98	98	98.98
[115]	MK-ELM	98.34	98.03	97.63	97.64
[204]	LEACH-ANN	97	96.8	96	96.4
[205]	CNN-MCL	99.46	99.76	99.15	99.46
[206]	HTM-LSTM	97.74	97.20	97.92	97.72
Proposed system	MLP-ANN	99.83	99.71	100.00	99.85

Figure 4. 15 (a) and Figure 4. 15 (b) Display the accuracy, precision, recall, and F1-measure in comparison to the performance of the proposed system using four benchmark datasets and several attack detection models. This provides evidence that the proposed scheme is useful for identifying and pinpointing the origin of attacks in WSNs. As can be shown in Figure 4. 15 (b), utilizing a sample standard benchmark datasets, ten-fold cross-validation, and three hidden layers, the proposed system has a higher average detection accuracy for routing attack detection and classification than the prior work by I. Almomani et al. [63]. To detect Sybil attacks, S. Dong et al. [3] employed the distance vector hop approach, achieving a localization accuracy of 78%, which is lower than the suggested scheme. If we look at the MK-ELM model [115],

which achieves an accuracy of 92.10% on the UNSW-NB 15 dataset, we can see that the proposed technique is also practically applicable.



(a) Comparison performance against benchmark datasets.

(b) Performance comparison of the proposed system with recent works.

Figure 4. 15. Evaluating the proposed system's attack detection and classification performance compared to existing based on benchmark datasets.

Figure 4. 15 compares the detection and localization capabilities of the proposed ANN approach to those of previous publications on Sybil attack detection. To evaluate the efficacy of Fuzzy Extreme machines, V.Sujatha and Anita [67] analyzed a sample experimental data set, finding an average detection rate of 97%. (FEM).

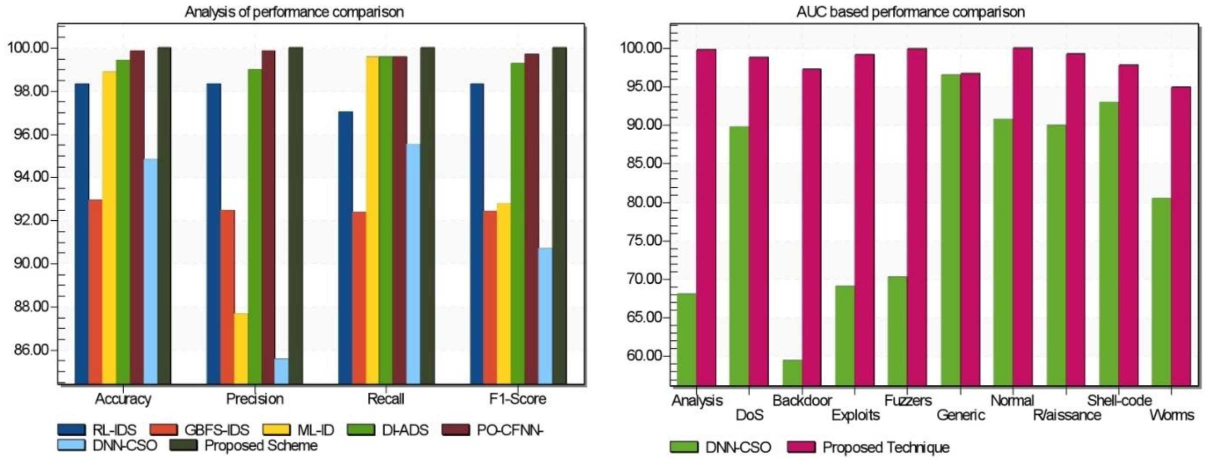
Using the same dataset, the suggested attack detection and localization approach achieves a perfect score of 100. Using the assessment criteria of packet delivery and energy consumption, B. Hasan et al. [207] concluded that an optimized artificial neural network could detect rogue nodes with an accuracy of 91.66%. From the results of these comparisons, we can infer that our suggested approach is superior at detecting and localizing attacks in WSNs, with the LEACH++ protocol built on an ANN (artificial neural network), F. A. Khan et al. [204] investigated the detection of routing attacks and found that they could be detected with an accuracy of 98%. That the suggested technique can detect routing assaults with an average detection accuracy of 99.62% is demonstrated. In addition, as can be shown in Figure 4. 15 (b), the suggested system achieves an average detection accuracy of 98.4% using the benchmark dataset NSL-KDD. With an average detection accuracy of 99.46%, the suggested method is useful for detecting and localizing DoS assaults in WSNs, outperforming the convolutional neural network and mean convolutional layer (CNN-MCL) model presented by L. Mohammadpour et al. [205]. As stated by W. Zhang et al. [115], Using the UNSW_NB and NSL-KDD benchmark datasets, a hierarchical intrusion detection model (HIDM) was proposed for WSNs with the help of a multi-kernel based extreme learning machine (MK-ELM) classification strategy.

By comparing the proposed system to prior studies with various types of attacks, we can validate its average localization and detection rate. Table 4. 5 demonstrates that the UNSW-NB15 dataset, used as a standard for identifying and classifying routing assaults, can enhance detection accuracy by class when employing 80% training and 20% testing of samples with five hidden layers. The demonstration also illustrates that the proposed performance parameters and metrics (accuracy, precision, F1-score, and recall) have been verified against those of previously published attack detection models.

Table 4. 5. Comparative analysis of the proposed technique with recent attack detection models using the UNSW-NB15 dataset.

Author	Method	Accuracy	Precision	Recall	F1-Score
A. M. Pasikhani,et al. [208]	RL-IDS	98.35	98.36	97.04	98.34
D. Upadhyay et al. [145]	GBFS-IDS	92.96	92.50	92.40	92.44
M. Abdan and H. Seno [158]	ML-ID	98.9	87.7	99.6	92.78
S. P. K. Gudla et al. [209]	DI-ADS	99.44	99.02	99.60	99.30
M. I. Alghamdi [49]	PO-CFNN-	99.86	99.89	99.58	99.72
R. Khilar et al. [210]	DNN-CSO	99.46	99.75	99.62	99.76
Proposed system	MLP-ANN	100.00	100.00	100.00	100.00

The proposed system is evaluated using benchmark datasets based on criteria including accuracy, precision, recall, and F1-score. This proves the system effectively detects and localises DoS assaults in WSNs, as in Figure 4. 16 (a).



(a) Comparison of the proposed system with recent works.

(b) AUC-based performance comparison using the UNSW-NB15 benchmark dataset.

Figure 4. 16. Examining the proposed system's performance of similar works using standard benchmarks.

The area under the curve (AUC) is also used for evaluating the system's performance, as shown in Figure 4. 16 (b). This demonstrates the efficiency of the optimized MPL-ANN method in detecting and localizing attacks on WSNs. Further comparisons are made between the proposed multilayer perception artificial neural network (MLPANN) method and MK-ELM on the NSL-KDD benchmark dataset, employing a subset of 14,000 sample records with three hidden layers, as shown in Table 4. 6 below. The suggested method outperforms MK-ELM, which uses 14,000 samples for its average detection accuracy of 98.34% due to its larger sample size of 111,110.

Table 4. 6. Comparison of performance of the proposed MLPANN with MK-ELM using the section of the NSL-KDD dataset with hidden layers.

	TP Rate (%)		FP Rate (%)		FN Rate (%)		TN Rate (%)	
Class of attacks	MK -ELM	Proposed scheme	MK-ELM	Proposed scheme	MK-ELM	Proposed scheme	MK-ELM	Proposed scheme
DoS	98.04	99.00	0.49	0.001	1.96	1.00	99.51	99.90
Probes	95.67	97.20	0.47	0.001	4.33	2.80	99.53	99.90
R2L	76.12	96.80	0.11	0.015	23.88	3.20	99.89	99.98
U2R	50.00	90.04	0.00	0.003	50.00	9.96	100.00	99.99

Theoretically and graphically, comparing the conclusion to prior works confirms its validity. We use the publicly available datasets UNSW NB, WSN-DS, and NSL-KDD to measure the efficacy of the multilayer perception artificial neural networks' (MLPANN) ability to recognize and classify a variety of threats. To improve the classification of machine learning models for the proposed scheme on the benchmark dataset, we combine a tree based on the Parzen estimation (PTE) with hyperparameter and Bayesian optimization (BO) techniques, as shown in Table 4. 7. Hyperparameters are used in every machine learning task to fine-tune the parameters above and achieve the best possible outcomes. As opposed to trial-and-error methods, hyperparameter optimization (HPO) uses machine learning to improve upon these outcomes while requiring less effort from the user [148].

Table 4. 7. Comparison of the performance of machine learning models using the CICIDS2017 benchmark dataset.

	ML Models results				Hybrid PTE-BO ML model results			
Classifier	Accuracy	Precision	Recall	F1Score	Accuracy	Precision	Recall	F1Score
XGBoost	99.82	99.82	99.82	99.80	99.82	99.86	99.82	99.83
Ensemble	99.82	99.91	99.82	99.85	99.82	99.91	99.82	99.85
RF	99.77	99.77	99.77	99.75	99.82	99.82	99.82	99.80

DT	99.77	99.77	99.77	99.75	99.82	99.91	99.82	99.85
ET	99.82	99.82	99.82	99.80	99.82	99.80	99.82	99.80

Compared to other methods on the WSN-DS benchmark dataset, ANNMLP's detection accuracy is higher, at 99.62%. The suggested approach efficiently pinpoints and identifies various types of assaults, validating the system's claim of optimal average detection for a large number of potentially malicious nodes. This work is novel since it successfully detects and localizes various attacks. The suggested technique is novel in wireless sensor networks with hierarchical architecture and heterogeneous and homogeneous sensor nodes because it can scale in security and performance for optimal area coverage.

4.9 Conclusion and Remarks

To identify and locate multiple assaults in WSNs, we propose a multilayer perceptron artificial neural network (MLPANN) in this work. When tested on the UNSW-NB, WSN-DS, NSL-KDD, and CICIDS2018 benchmark datasets, respectively, the proposed scheme's average detection accuracy for the various malicious nodes was 100%, 99.65%, 98.95%, and 99.83%. With an average localization accuracy of 99.46% employing 160 beacon nodes, the optimized localization approach is more successful and performs considerably by 20% compared to the distance vector hop technique. Previous research utilizing the ANN classification methodology with Python, IBM SPSS, and WEKA Toolboxes for data processing and MATLAB R2021a for network planning and simulation confirms the validity of the suggested method. The proposed system's detection and localization accuracy for various assaults is measured using the datasets. Metrics such as detection rate, ROC, false-positive rate, network lifetime, residual energy, and area under the curve are used to evaluate the performance of the suggested method. We Simulated the intended field with a beacon, sensor, and malicious nodes. Several ways are suggested to improve the accuracy of hierarchically WSNs' detection and localization of malicious nodes. In the future, we plan to add new types and techniques of attacks to this effort. The findings demonstrate that the suggested scheme's performance and security may be applied to scalable and extensive network coverage in wireless sensor networks with heterogeneous and homogeneous sensors to guarantee service quality and availability. In the future, other network plans and techniques will be used alongside other public datasets to serve as standards against which the suggested scheme for detecting and localizing assaults in WSNs can be measured.

Chapter 5

5 BLOCKCHAIN-ENABLED SECURE LOCALIZATION BASED ON FEDERATED LEARNING

This chapter addresses the challenges posed by malicious nodes in WSNs and proposes a solution that leverages the decentralized nature of blockchain and the collaborative learning approach of federated learning. Here are the main objectives and contributions discussed in this chapter:

- By incorporating blockchain technology, the chapter aims to enhance the security and trustworthiness of the localization process to ensure that the localization results are accurate and reliable.
- To detect and mitigate the presence of malicious nodes in the WSN. Malicious nodes can compromise the accuracy and integrity of the localization process by injecting false or misleading information.
- To leverage federated learning as a collaborative approach to enhance localisation accuracy in WSNs. Federated learning enables distributed nodes to train a machine learning model collaboratively without sharing sensitive data.
- Incorporating blockchain technology is to establish trust and transparency in the localization process. Blockchain provides a decentralized and tamper-resistant platform for storing location information, consensus mechanisms, and smart contracts.
- To evaluate the performance of the proposed blockchain-based secure localization system. This involves conducting experiments, simulations, or real-world deployments to assess the system's accuracy, efficiency, and scalability.

5.1 Introduction

Several significant societal and economic shifts have resulted from the proliferation of the Internet of Things (IoT), which has been made possible by recent developments in technology and network infrastructure [16]. The Internet of Things (IoT) refers to the global infrastructure of interconnected computing equipment, software, and humans that facilitates data processing

[17]. The military, smart cities, healthcare, and the environment are just a few fields that have recently benefited from the Internet of Things-based wireless sensor networks (IoT-WSNs) [19]. Wireless Sensor Networks (WSNs) have both practical and academic significance and have contributed significantly to the rapid expansion of the Internet of Things (IoT). Wireless sensor networks (WSNs) are ad hoc networks comprising a swarm of devices, or "sensor nodes," all outfitted with sensing technologies. Self-organizing, randomly deployable, fault-tolerant sensor nodes that closely monitor a large area are essential. Internet of Things-Wide Area Networks are parameter-variable, self-organizing networks for environmental monitoring and data collection. The sensor nodes communicate and assess one another so that data may be transmitted efficiently and malicious nodes can be identified and removed from the network. By compromising the beacon node in the cluster area, hostile attackers can exploit information and generate erroneous routing information between the sensor node and the base station. Through false information broadcasting and the subsequent erosion of confidence between nodes and end users, the malicious node reduces the efficiency of the network. A network's sensor nodes monitor threshold and signal intensity readings to identify and analyse possibly malicious nodes as part of its trust evaluation process.

Blockchain, a distributed ledger technology, ensures the security of Bitcoin networks [20]. The innovative architecture is widely used in many distributed settings, such as healthcare and automotive Adhoc networks (VANET). Several studies have found that blockchain technology is particularly well-suited to tackling security concerns related to the Internet of Things (IoT) for three main reasons. A blockchain network operates similarly to an IoT network in that all network nodes store and processes the network's data collectivelyDistributed blockchain technology has various advantages over traditional central administration methods for keeping a large number of IoT devices online and working at a cheap cost. Many distrust IoT gadgets because of their short battery lives. This means that personal data on these devices is vulnerable to theft or exposure. Blockchain's unparalleled immutability eliminates the need for trusted parties to transmit data in a way that respects their respective privacy and security. Using a hash, each block in the blockchain stores a copy of the information from the prior block. Many scenarios call for the gadgets to be entirely untraceable. However, most devices have a simple architecture that can't handle the complexity of typical encryption techniques. With the blockchain design of IoT-WSNs, entities interact with one another through transactions as the primary unit of data exchange. All records of a customer's financial dealings are called a "transaction" in the blockchain's introductory form.

Deep Learning (DL), a subfield of Machine Learning (ML), has made tremendous strides in recent years and has applications in many fields. Additionally, ML includes shallow models [211]. As a result, it can offer a way to spot some types of attacks without requiring much human involvement. ML-based intrusion detection systems (IDS) can self-train by comparing "normal" and "abnormal" traffic patterns. As such, it can be used to identify potentially dangerous traffic patterns, either within a dataset or in the wild. Although these methods have proven useful for IDS, they often require a centralized administration to process data from all users on the network. A high packet loss rate lowers prediction accuracy as network capacity increases. In this research, we present a hybrid federated machine-learning strategy for tracking down and tracing attacks in IoT-WSNs by leveraging the immutable and decentralized ledger technology blockchain.

5.1.1 Blockchain Applications in IoT-WSNs

Blockchain is a distributed database that stores transaction records and ensures that all nodes in the network have the same cryptographically secure copy of the data. Bargaining data is stored in a chain of blocks, each with its hash value, and may only be used to record a certain number of transactions. Due to the hash values in each block, the information on the blockchain cannot be altered [20]. When most nodes in a network verify a transaction, that transaction is added to the blockchain, which is an immutable distributed ledger. The phrase "distributed ledger" characterizes the decentralized data storage system it depicts [72]. The network consensus is required for every ledger transaction to be considered legitimate. Bitcoin is arguably the most well-known real-world implementation of blockchain technology. The benefits of integrating blockchain technology and the IoT are numerous. To illustrate, a distributed ledger system that is based on the blockchain can be used to coordinate IoT gadgets and send up-to-date information to each node in the network in real-time. Unfortunately, these apps generate vast amounts of data, which might cause issues related to big data.

Consequently, Artificial Intelligence (AI) works as an analytical tool and contributes to decision-making, categorization, prediction, and detection in a blockchain-enabled IoT network to solve this challenge. Moreover, the analysis is carried out in the cloud under the present blockchain-based architecture. Yet there are drawbacks to using a centralized server, such as low speeds and accuracy, low latency, and restricted computational storage. In order to determine whether or whether the new distributed paradigm of "fog computing" can fulfil these requirements, it needs to be investigated in depth. Cisco first used the term "fog computing" in 2012 to describe a distributed architecture that places cloud services in the network's outer reaches. Many aspects of blockchain-based systems can benefit greatly from adhering to fog computing concepts, such as location awareness, support for heterogeneity, low latency, mobility, and geo-distribution. Fog computing has found use outside of load balancing and data collection. To counter distributed denial-of-service (DDoS) assaults on mining pools in a blockchain-enabled Internet of Things (IoT) network, this study blends fog computing and artificial intelligence. This diagram shows how blockchain, the Internet of Things, artificial intelligence, and fog computing may work together to form a safe attack detection model, as in Figure 5. 1 for various applications.

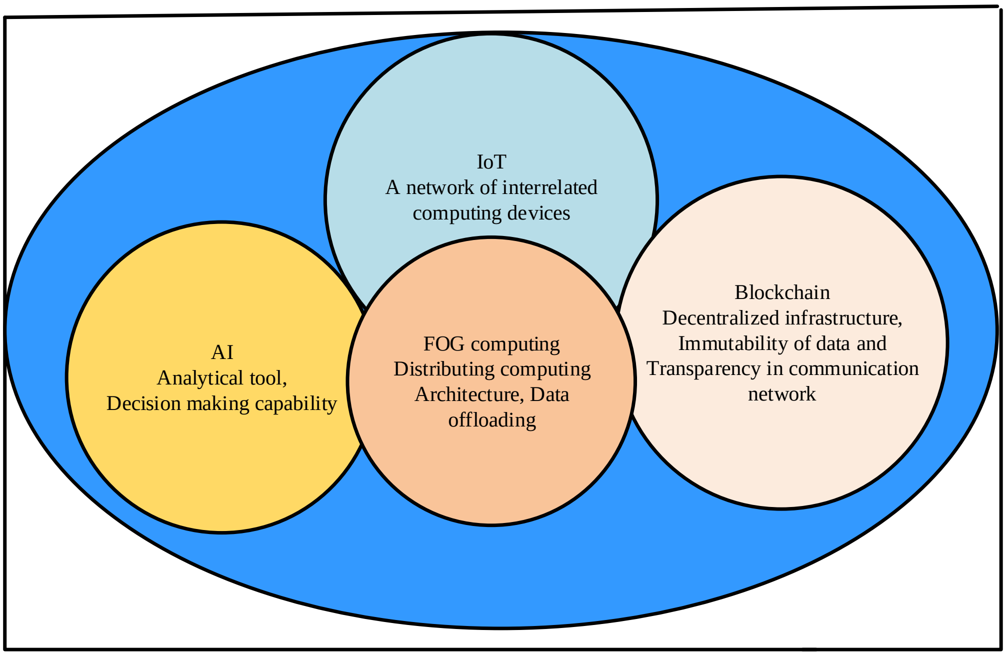


Figure 5. 1. Illustrating secure IoT-based WSNs showing a detection model incorporating various application domains, including Blockchain, IoT, AI, and Fog computing [7].

Blockchain is a promising new technology that is already seeing widespread use. Its widespread adoption is due to the security and distributed trust it provides amongst peers in many settings [212]. Integrating blockchain technology with IoT allowed for stronger authentication and authorization in WSNs that rely on IoT. To address this issue, we offer a decentralized localization method built on the blockchain and a pseudonymous permission management framework that grants users full control over their data while keeping it secure. The models based on blockchain technology are distributed authorization databases. To better understand how blockchain technology can be applied to IoT based-WSNs, a new research topic has emerged [213].

Some researchers have begun implementing blockchain technology in WSNs to address data processing, storage, node failure, and user access control issues. In Mobile Edge Computing (MEC), the server can save encrypted block hashes in memory and send mining tasks that are computationally intensive to the edge nodes. Using census techniques, proof of work (PoW) and proof of state (PoS), and proof of authority (PoA), the blockchain can reach a consensus between previously untrusted nodes (PoA). The blockchain verifies the integrity of the network by recording each transaction in a chronological block with associated headers and bodies. The headers are the transaction's root, the preceding block's hash, and the timestamp. In IoT-based wireless sensor networks, blockchain techniques are deployed across domains to precisely pinpoint individual beacon nodes' locations. Via their senses, sensor nodes in a network relay information about their precise location and identity to a designated "sink" node. As a result of beacon nodes being compromised and false information being sent to the sink node, malicious nodes might impact the location and position of sensor nodes. The client nodes can depend on the blockchain-based secure localization system because it uses a non-repudiation concept.

A sensor node's precise geolocation can be ascertained using a global positioning system (GPS) or another method [214]. One strategy is to use mathematical approaches to pinpoint the precise positions of individual nodes and then centrally handle this data. The Support Vector Machine is widely deployed in WSN to verify nodes' identities and protect sensitive data. For mobile node location, we employ support vector machines and data on network availability. The received signal strength indicator (RSSI) is used to pinpoint the node's location. Localized SVM allows for fast and widespread localization, although it is still subject to outliers in the training data. Localization accuracy has improved thanks to Support Vector Regression (SVR), an innovative data extraction approach for localization models. Improves localization accuracy without significantly raising equipment costs. The unknown node's location can be precisely determined using SVM-based learning with limited anchor nodes. The accuracy of the grid is ensured by its discrete cells. An SVM-based DV-hop algorithm performs admirably in large-scale networks. Sensor nodes have access to some AI capabilities (SOM) by employing a localization technique based on a Self-Organizing Map. Without any guidance from a person, SOM can automatically learn to classify data. Using a Self-Organizing Map, the precise location of each sensor node is determined. In this case, the input layer comprises the locations of eight anchor nodes close to the mystery node. After the network has been trained, its output layer will reveal the 2D coordinates of the unknown node. The disadvantage of this method is that it requires nodes to be evenly spaced throughout the monitored region.

This research offers a distributed intrusion detection system (IDS) built on fog computing for use by mining pools in blockchain-enabled IoT networks. The IDS can secure planning and simulation, attack detection, and categorization. The proposed IDS is trained with the help of hybrid federated machine learning algorithms. An overarching perspective on the proposed decentralized structure. The detection system's foundation consists of sensing nodes, each responsible for keeping tabs on and labelling any moving objects that enter its field of view. All IoT sensors are classified into one of several categories based on their results during training and testing, and all of these categories have the same detecting capabilities. This group shares its information with adjacent fog nodes. Since fog nodes serve as an entry point or gateway for IoT gadgets, they must implement an intrusion detection mechanism to safeguard the network. Intrusion detection systems (IDS) examine incoming traffic and take appropriate measures based on their findings. In the event of a typical data influx, the transaction is announced to the entire mining pool's storage. The miners in a blockchain network in the cloud select which transactions to mine and generate new blocks. If malicious or invalid transactions are detected, the IDS will trigger an alarm, allowing the administrator to take appropriate action. The proposed method has two main benefits: it can detect routing risks in IoT-based WSNs at scale and has excellent performance for detecting such threats. It performs admirably when integrated into an IoT system that uses many wireless sensor networks.

5.1.2 Blockchain-Based Secure Localization

We presented a blockchain-based safe localization method that uses a trust evaluation mechanism and cascading encryption to effectively identify rogue nodes and ensure all network transactions' integrity. Since malicious nodes impact the accuracy of the localization approach in IoT-based wireless sensor networks, the randomly placed sensor nodes suffer security concerns during the localization process. This issue can be solved because of the blockchain's cascade encryption and the beacon nodes' efficient transaction and trust evaluation procedure. The blockchain receives a list of trusted nodes through distributed consensus and can identify any malicious actors within that group.

The position of the nodes interrupts the normal operation of the network and provides data to the cluster head without the cluster head knowing where those nodes are located. Beacon nodes, which know exactly where they are, can use this information to pinpoint the precise location of any other nodes in their deployment before delivering any data, thus avoiding this issue. Calculating the trust value of each beacon separately from the malicious node allows for determining which beacon nodes can be trusted. Blockchain-based localization solutions based on federated learning can detect the location of the rogue node without compromising the beacon node or users' privacy. The suggested approach divides the country into networks, each with many clusters dedicated to tracking the local environment. The network's nodes, when working together, can achieve their aims; as a result, the interactions between them must demonstrate the validity of each node's claimed identity [215]. Authentication of identities is an important part of keeping the Internet of Things safe. Two-factor authentication is required to establish encrypted channels between nodes in the described network model before any communication can take place, or end users may have access to the resources hosted by a node.

5.1.3 IoT-Based Network Model

The model network comprises ten distinct regions, each containing a unique assortment of wireless sensor nodes performing a unique computational task. As can be seen, each area is broken up into smaller sub-areas called clusters, as in Figure 5. 2. Based on their processing capacity and role in the network, nodes are categorized as sinks, cluster leaders, or sensors. It is assumed that the sensor nodes are dispersed randomly and have no idea where they are. While the beacon and sink nodes know their placements, the unknown sensor nodes rely on their knowledge of the distance between each node and its position to locate the sensor nodes and identify the malicious nodes [32]. Several design constraints and assumptions were incorporated into the network model to ensure optimal performance. Among the prerequisites are:

- The topology is considered to change with the mobility of randomly placed sensor nodes. In the IoT, each gadget has its unique IP address.
- The capacity of the cluster's head nodes and base stations, which can be used to execute a smart contract, are detailed.
- A persistent malicious node in the network causes the architecture to be hypervigilant.

- Every single one of the sink nodes can be relied on to create keys for the rest of the network reliably. All sink nodes are equipped to the hilt with powerful computing and data processing capabilities regarding data aggregation.
- Data in the network is safely and reliably stored on the servers.

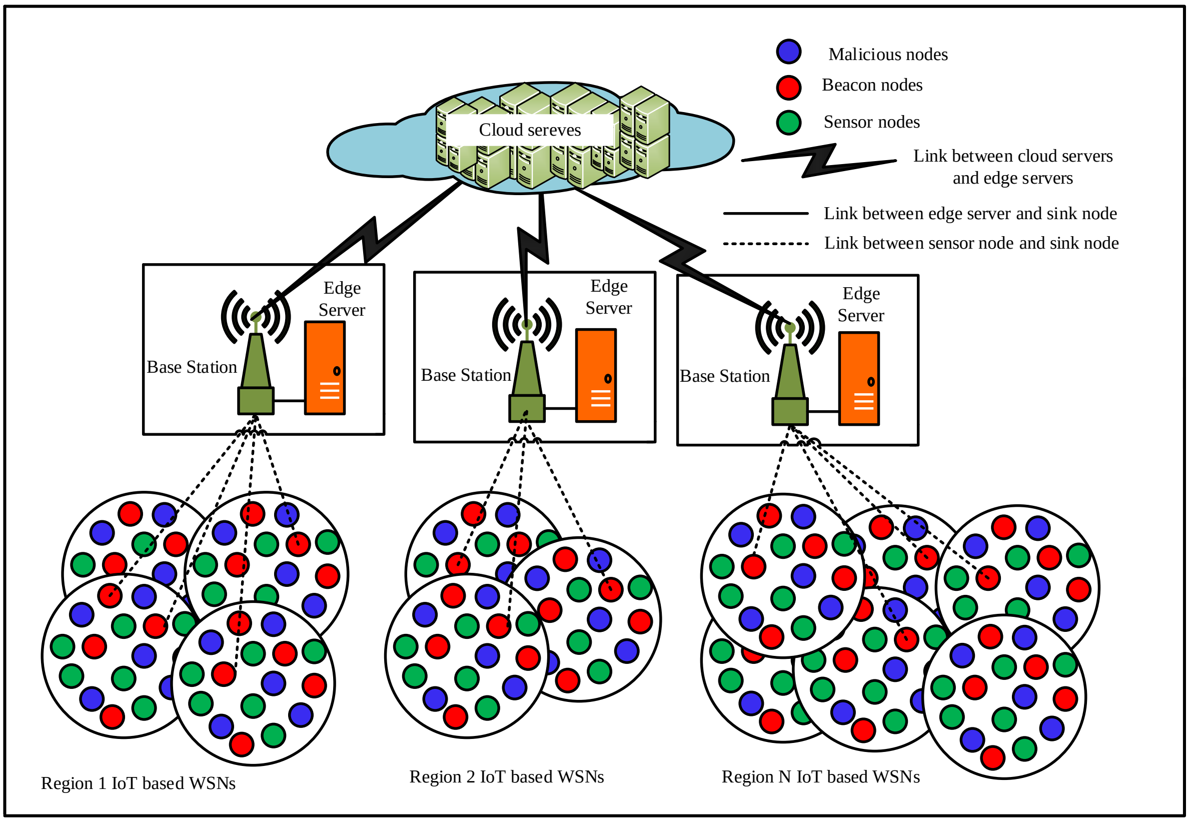


Figure 5. 2. Hierarchical model of an Internet of Things–based wireless sensor network proposed for the secure localization of malicious nodes.

IoT-based wireless sensor networks have two challenges: hostile information dissemination through node takeover and inaccurate data regarding energy consumption and physical location, both of which compromise the networks' ability to provide accurate localization services.

The suggested system has N sensor nodes, B beacon nodes, and U unknown nodes, all randomly dispersed in a two-dimensional environment. Each region's base stations are placed beyond the clusters' edge server connections. The equations (5.1) are used to determine the beacon's network size and communication radius shown below.

$$|N| = |B| + |U| \tag{5.1}$$

The transmission range (T_R) of the beacon nodes is computed with the maximum (T_{max}) and minimum (T_{min}) communication ranges as shown below in equation (5.2).

$$T_R = (T_{max} - 1) + \text{random}(0,1) \times \left[(T_{min} - 1) - (T_{max} - 1)\right] + 1 \tag{5.2}$$

Unknown nodes in a wireless sensor network can pose a security risk. Hence the network must constantly estimate their location and pinpoint their precise location. Accurate localization in beacon-based WSNs can be achieved by detecting malicious sensor nodes, which provide data for calculating location.

5.2 Problem Statement

Wireless Sensor Networks (WSNs) are vital in various Internet of Things (IoT) applications, enabling efficient data collection and communication. However, malicious nodes within WSNs pose significant security risks, particularly in terms of localization accuracy. Traditional localization strategies in WSNs are often centralized and vulnerable to attacks, leading to localization errors and compromised data integrity. Here are several challenges and solutions to be addressed:

- Current solutions often focus on individual components, such as blockchain-based trust administration or federated learning-based machine learning models. There is a need for a comprehensive system that integrates these components seamlessly to provide robust and secure localization against malicious nodes.
- Feature evaluation plays a crucial role in detecting and classifying malicious nodes accurately. Existing approaches may not effectively evaluate the features extracted from WSNs, leading to suboptimal performance in localization. Improved techniques are required to identify relevant features that can enhance the accuracy of the localization process.
- The localization process in WSNs relies on the accuracy of information provided by beacon nodes. Malicious nodes can inject false location data, leading to incorrect localization. Ensuring the reliability and trustworthiness of beacon nodes while considering the presence of malicious nodes is a significant challenge that needs to be addressed.
- As WSNs can consist of a large number of sensor nodes, scalability and efficiency are critical factors for any proposed solution. The system should be able to handle the computational and communication overheads associated with blockchain-based secure localization while maintaining real-time performance.

In light of these challenges, there is a need to develop a blockchain-based secure localization system that leverages federated learning techniques for accurately detecting and localising malicious nodes in IoT-based WSNs. This system should address the limitations of existing approaches and provide robust localization while ensuring data integrity, reliability, scalability, and efficiency.

5.3 Research Contribution

A wide range of challenges and worries about the security of IDS, detection, and classification attacks using blockchain technology in IoT-based wireless sensor networks are revealed in the literature. All IoT networks backed by blockchain will be at greater risk as distributed denial of service (DDoS) attacks rise in the blockchain IoT ecosystem. Defending a blockchain-powered IoT network calls for a sophisticated distributed security framework. Ensuring a decentralized security system applies the appropriate analytical techniques to the large amounts of data produced by IoT devices. Making an IDS that can distinguish between safe and harmful Internet transactions is challenging. Once the corresponding model has been developed for blockchain-enabled IoT wireless sensor networks, little is known about preventing distributed denial of service (DDoS) attacks against mining pools. To solve this problem, we analyzed relevant datasets to provide a hybrid federated machine learning-based technique for identifying and localizing routing attacks in WSNs-IoT.

In light of these issues, we proposed implementing Federated Learning (FL)-enabled secure blockchains in IoT wireless sensor networks to detect and pinpoint the precise locations of attacks. Using FL, devices can collaborate to learn without sharing data with a central server. That is to say that ML/DL can be trained in a distributed manner, with various devices and servers processing data across iterative training. Local learning and model transmission are the two key pillars of this strategy, and they allow for the same privacy protection and cost savings as more conventional centralized machine learning methods. The complete ML/DL model can be improved using FL over time. The FL server selects a subset of clients at the beginning of each round to facilitate learning and provides them with the most up-to-date global model. The advantages of the blockchain-based secure localization that we propose to use in IoT-based wireless sensor networks are listed below:

1) Exploring blockchain techniques for detecting and localizing malicious nodes in IoT-based wireless sensor networks involves studying how blockchain can be applied to enhance security in the network. By leveraging the blockchain's decentralized and immutable nature, detecting and localizing malicious nodes effectively becomes possible.

2) Enhancing the detection and localization of malicious nodes in IoT-based wireless sensor networks aims to improve service quality by identifying and mitigating potential security threats. By implementing advanced detection algorithms, anomaly detection mechanisms, and localization techniques, it becomes possible to identify and isolate malicious nodes more accurately.

3) Exploring the effectiveness of federated learning techniques using performance metrics allows for evaluating the attack detection performance in IoT-based wireless sensor networks. Federated learning is a distributed learning approach that enables collaborative training of machine learning models using data from multiple devices or nodes. By applying federated learning to detect and mitigate attacks, performance metrics can be used to assess the attack detection mechanism's accuracy, efficiency, and effectiveness.

4) Providing secure routing and data transmission in a hierarchical IoT-based wireless sensor network is crucial for ensuring secure service provisioning. The network can establish certain communication channels and protect sensitive data during transmission by employing encryption, authentication, and secure routing protocols. Hierarchical structures enable the division of the network into multiple levels, allowing for efficient routing and centralized control of security measures.

5) Achieving effective detection and localization of malicious nodes by computing the unknown nodes based on the blockchain technique with the help of beacon nodes involves utilizing blockchain technology to identify and localize malicious nodes accurately. Beacon nodes, which act as reference points in the network, can assist in the localization process by providing trusted information. Unknown nodes can be computed and verified by leveraging the blockchain technique. This strategy effectively detects and localises malicious nodes in IoT-based wireless sensor networks.

6) Ensuring secure data aggregation and trust value computation of beacon nodes using a trust management model based on blockchain enables the removal of malicious nodes and ensures reliable service provision. By utilizing blockchain-based trust management, the behaviour and trustworthiness of beacon nodes can be evaluated and verified.

7) Designing and planning secure range-free localization processing using blockchain technology-based selection of trust value of the miner beacon nodes involves utilizing

blockchain for secure localization in range-free techniques. By incorporating the trust values of miner beacon nodes stored in the blockchain, the selection process for localization can be enhanced.

8) Exploring blockchain-based secure data aggregation and localization of unknown nodes for malicious node detection and localization using federated learning approaches involves combining blockchain and federated learning techniques for improved security. By leveraging the blockchain for secure data aggregation and localization, along with federated learning approaches, the network can achieve accurate detection and localization of malicious nodes.

5.4 Proposed System

With the help of hybrid and federated learning-based blockchain technology, the suggested system for detecting and localizing rogue nodes in IoT-based wireless sensor networks comprises multiple steps for recognizing and removing threats, as shown in Figure 5. 3. Effective transaction and localization are guaranteed by the proposed scheme's use of hybrid machine learning and federated learning models, feature assessment, and cascade encryption. By preparing the data received from the blockchain's trust value lists into an appropriate format, the suggested solution used a federated machine learning model to discover and classify rogue nodes. It can be challenging to ensure the trustworthiness of beacon nodes in a WSN, whose information is crucial to the localization process. However, when malicious nodes are present in an IoT-based WSN, the range-free localization method can fail due to its lack of reliance on specialized hardware. The malicious nodes provide False location data during the localization process [216]. Common localization approaches for WSNs have a single point of failure since they rely on a coordinating entity. Trust evaluation based on a secure blockchain should be tested for attack detection and localization to see how well it works with hybrid federated machine learning, reinforcement learning, and maximum likelihood estimation. The blockchain has potential uses in the area of trust management. In addition, only a limited number of factors are used to assess whether or not to trust a beacon node. Because of this, the ultimate trustworthiness judgement must consider the integrity of both behaviours and data. From there, the blockchain infrastructure receives the data from the trusted nodes and uses it to do its thing.

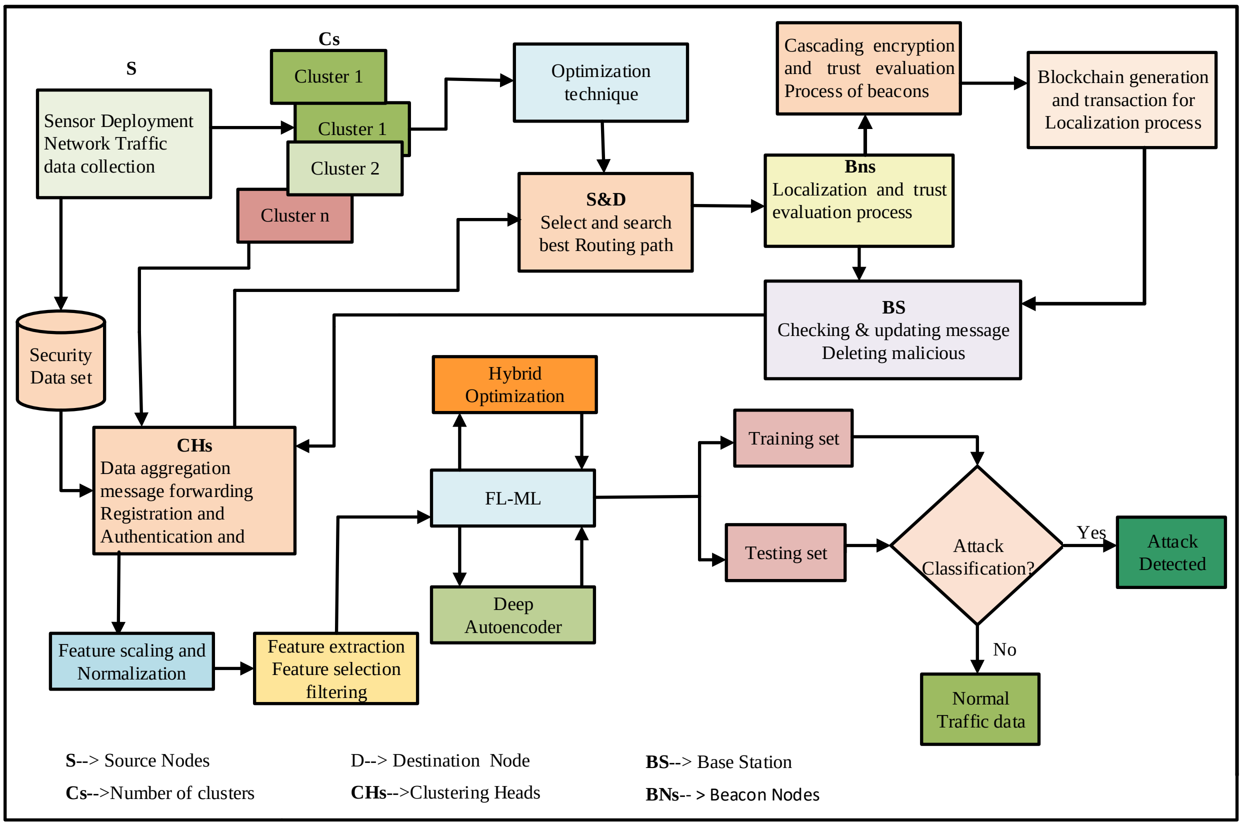

S
Cs
Sensor Deployment Network Traffic data collection
Cluster 1
Cluster 1
Cluster 2
Cluster n
Optimization technique
Cascading encryption and trust evaluation Process of beacons
Blockchain generation and transaction for Localization process
S&D
Select and search best Routing path
Bns
Localization and trust evaluation process
BS
Checking & updating message Deleting malicious
Security Data set
CHs
Data aggregation message forwarding Registration and Authentication and
Hybrid Optimization
FL-ML
Deep Autoencoder
Training set
Testing set
Attack Classification?
Yes
No
Attack Detected
Normal Traffic data
Feature scaling and Normalization
Feature extraction Feature selection filtering
S--> Source Nodes
Cs-->Number of clusters
D--> Destination Node
CHs-->Clustering Heads
BS--> Base Station
BNs-- > Beacon Nodes

Figure 5. 3. A block diagram showing the proposed blockchain-based attack localization and detection strategy for wireless sensor networks (WSNs).

The proposed system aims to overcome the limitations of traditional localization strategies in WSNs, which often rely on a centralized body and can be susceptible to attacks. By leveraging hybrid and federated learning-based blockchain technology, the system enhances the detection and localization of malicious nodes in IoT-based WSNs.

5.4.1 Minor Node Selection and Localization Process

Beacon nodes in WSNs use several types of trust computation methods, including those for computing behavioural, data-based, feedback, and overall trust. After the trustworthiness of each node is determined, the most trustworthy beacon nodes are selected. These beacon nodes are reliable for mining and triangulation. These miners form the backbone of the Proof-of-Authority consensus that underpins our blockchain network. This network uses a private blockchain on the sink and beacon nodes. The nodes that are reliable beacons are recorded forever on the blockchain. The nodes in the network also encrypt the transaction data before adding it to the distributed ledger. The miner node acts as the most reliable central hub, storing the most important data used to validate communications and create new blocks for processing. The translation procedure will then start after this. Each unknown node's specific location can be found using trilateration. Unknown nodes here estimate their location by finding the average distance between themselves and the three anchor nodes with the greatest trust ratings and then using that information to solve for their coordinates using equation (5.3)

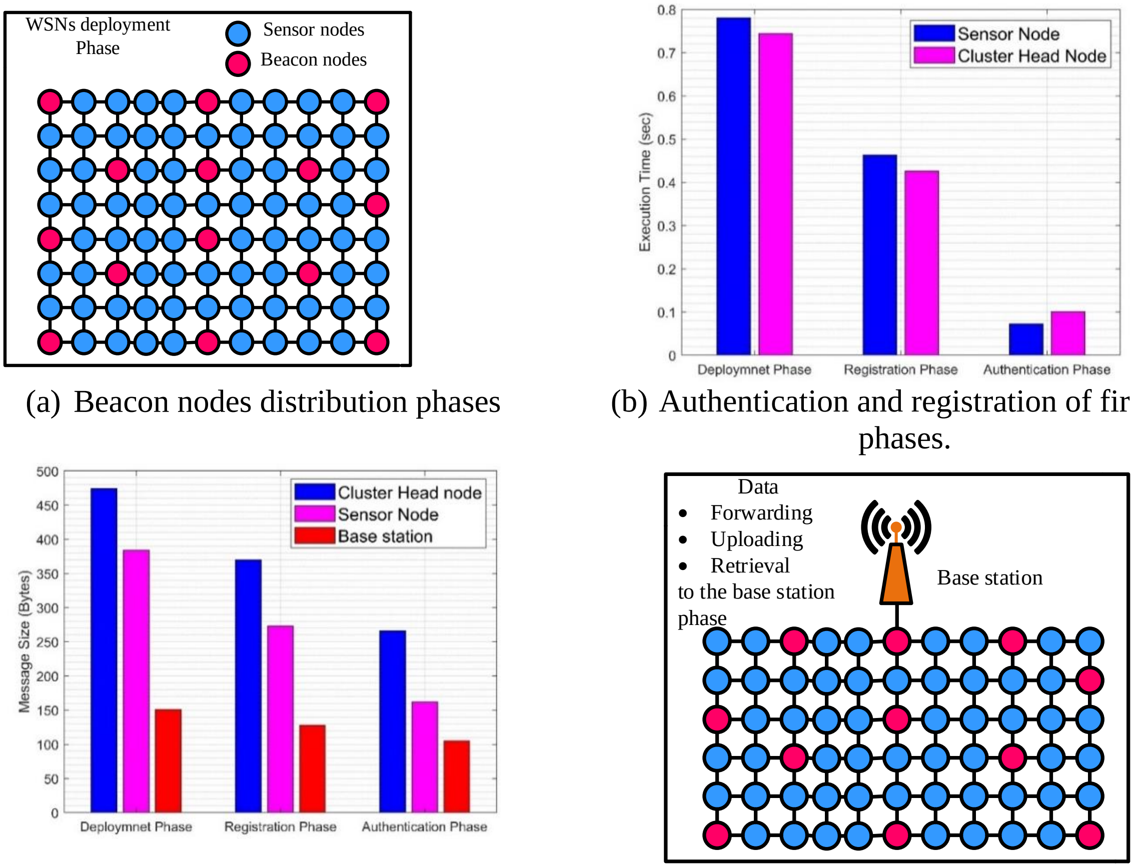
$$D_{tri} = \begin{cases} \sqrt{(u_a\text{-}u_i)^2 + (v_a\text{-}v_i)^2} \\ \sqrt{(u_a\text{-}u_j)^2 + (v_a\text{-}v_j)^2} \\ \sqrt{(u_a\text{-}u_k)^2 + (v_a\text{-}v_k)^2} \end{cases} \tag{5.3}$$

Where D_{tri} is the trilateration process for finding the distance of any node a (u_a, v_a), (u_i, v_i), (u_j, v_j) and (u_k, v_k) are the positions of the ordinary sensor nodes concerning i, j, and k beacon nodes.

Unique methods are necessary to evaluate the position of the nodes and their connections and determine where they are located. The system is divided into two groups: those that rely on range and those that don't [32]. Although the second choice is more economically viable, it requires specialist gear. The received signal strength indicator (RSSI) and the distance vector hop localization algorithms are used to assess the precision and location of wireless sensor nodes. A distance vector localization mechanism is needed to ascertain the locations of sensor nodes and cluster leaders in relation to beacon (base station) nodes [10]. The strategy can be used to determine the unknown nodes' position and adjust their distance from one another. In order to pinpoint beacon nodes in a WSN, the distance vector hop method is typically employed. The minimum distance is calculated using the average hop size in the distance vector method [3].

5.4.2 Registration and Authentication Process

The three primary types of sensor nodes in the proposed network paradigm are sensor nodes, aggregation nodes, and base stations [74]. Through its Media Access Control (MAC) address and the base station's authentication procedures, intelligent communications authenticate the existence of the aggregation and data processing node, as in Figure 5. 4. The sensor nodes require authentication and authorization before they can access the model and the used public blockchain [217]. For this reason, our model employs a layered security architecture to keep the nodes' data secure. Establishing trust between different IoT participants is crucial for the proposed IoT-WSNs framework's ability to detect and localize threats. Figure 5. 4 (a) illustrates wireless sensor network node deployments.



(a) Beacon nodes distribution phases

(b) Authentication and registration of first phases.

(c) Registration and authentication of sensor nodes in WSNs.

(d) Data retrieval and uploading phases

Figure 5. 4. Authentication, registration, and transmission procedures involved in secure data collection and transfer using IoT-WSNs.

In Figure 5. 4 (b) and Figure 5. 4 (c), During deployment, registration, and authentication, we can see that the cluster head (CHs) collects a huge message size while the sensor nodes (SNs) spend an outsized amount of time performing data execution. The cluster leaders with position and location awareness are called beacon nodes. Information is transmitted, retrieved, processed, and consolidated at the base station, as depicted in Figure 5. 4(d).

Authentication methods in WSNs can be trusted because of the combined data from all of the nodes and the public blockchain records of verified nodes. Once the registration process is complete, sensor nodes are granted access to the blockchain, thus strengthening the security of WSNs against external attacks. Sensor nodes, initially spread in a random pattern across the playing field, later coalesce into larger nodes known as aggregation nodes. The aggregation nodes authenticate the sensor nodes through a private blockchain during communication, while the base station uses a public blockchain to confirm the identity of the aggregation node. All communications between nodes in the aggregate are secured with mutual authentication.

5.4.3 Federated Learning-Based on Blockchain Technique

To identify and pinpoint malicious nodes, the suggested system, as shown in Figure 5. 3, uses hybrid and federated learning-based blockchain technology to detect and eliminate assaults in IoT-based wireless sensor networks throughout several phases. The proposed approach uses hybrid machine federated machine learning models, feature evaluation, and cascade encryption to guarantee the provisioning of clients and services for efficient transaction and localization. By training and testing the dataset obtained from the Blockchain's trust value lists in a suitable data format using preprocessing techniques, the proposed system used a federated machine learning model to detect and categorize rogue nodes. However, since the localization process in a WSN depends on the accuracy of the information relayed by beacon nodes, it can be challenging to ensure their dependability [216]. Due to the lack of reliance on specialist hardware, the accuracy of range-free localization in IoT-based WSNs can suffer in the presence of malicious assaults.

Regarding localization, the malicious nodes lie and provide inaccurate positions. Many localization approaches for WSNs have a single point of failure since they rely on a coordinating entity. Trust evaluation based on secure blockchain employing hybrid federated machine learning, reinforcement learning, and maximum likelihood estimation must be tested to determine its effectiveness in detecting and localizing attacks. The blockchain has potential uses in the area of trust management. As an added caveat, only a limited number of criteria are used to assess whether or not to trust a beacon node. Because of this, the ultimate trustworthiness judgment must consider the integrity of both behaviour and data. The data is then collected and processed by the blockchain infrastructure provided by the trusted nodes.

Recent advances in Machine Learning (ML) approaches, specifically Deep Learning (DL) models, have improved several sectors, including medicine, computer vision, and wireless communication. Deep models aren't the only type of ML models out there [211]. There may be ways to detect certain attacks without much human involvement. ML-based intrusion detection systems (IDSs) can self-train by comparing "normal" and "abnormal" traffic flows. It can be used to spot potentially dangerous traffic patterns, whether they exist in a dataset or the real world. While these methods have proven useful for IDS, they often require a centralized administration to collect and aggregate data from all users on the network. It has been discovered that larger network sizes are associated with decreased prediction accuracy when packet loss rates are high. As part of this study, we propose employing blockchain, an immutable and distributed ledger technology, to trace malicious activity in IoT-WSNs back to its initial entry point.

The current standard for providing security for Internet of Things (IoT) applications is the federated cybersecurity (FC) paradigm, which relies on communication and cooperation between multiple parties. However, the conventional approach to data transmission within the cluster and between groups presents privacy and security problems. This has led to the development of federated learning (FL). This approach allows for the safe and confidential transfer of data and knowledge, as in Figure 5. 5. In order to keep the IoT network safe. Using an FC model that collaborates with FL is necessary to communicate and exchange information at any level.

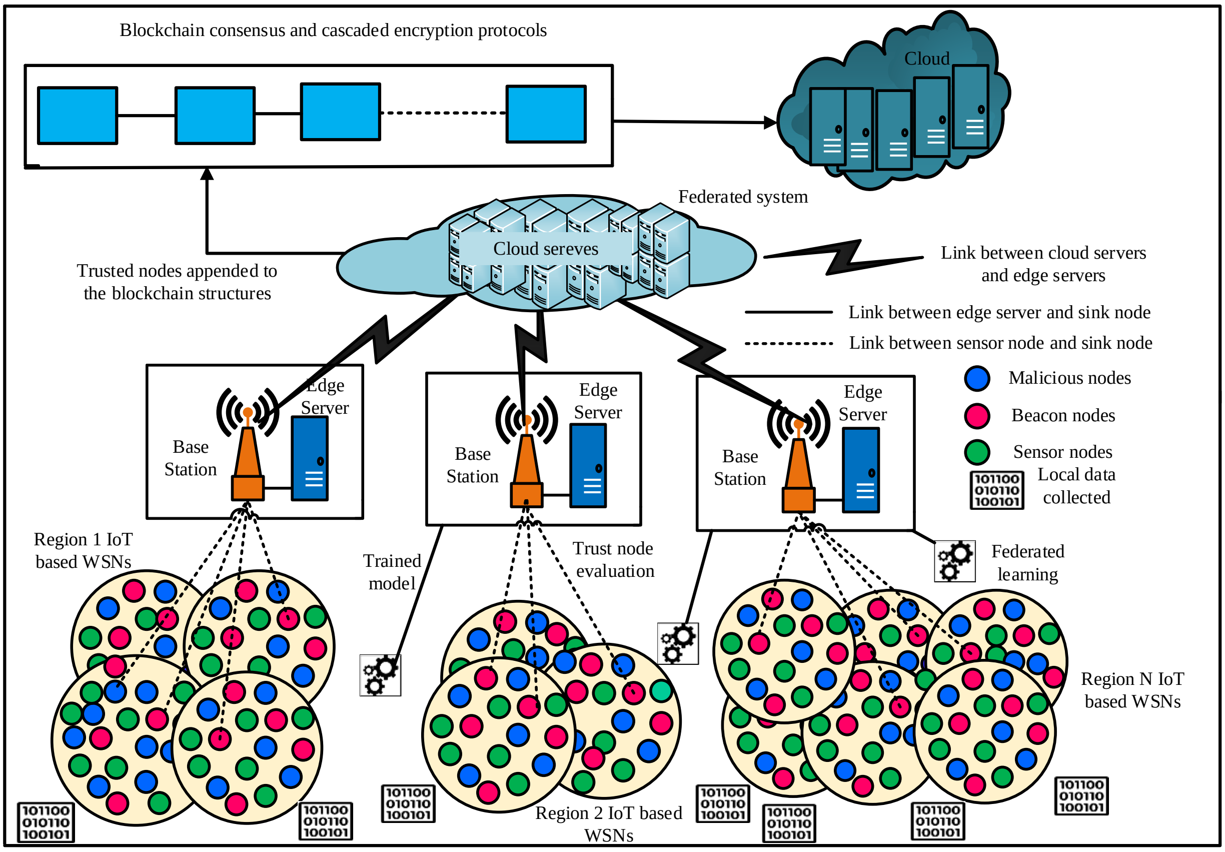


Figure 5. 5. Attack detection and localization in a secure hybrid federated Internet of Things-based WSNs model.

Regarding cyber protection, most FC plans based on FL have focused on safeguarding IoT networks under the assumption of a single, global service offering. While the method was designed for a lone global model under the control of a single service provider, it may easily be extended to a more complex collaborative scenario, including multiple global models under the control of multiple service providers. There hasn't been a lot of work put into creating a federated security model that considers various international methods [218]. In IoT-based WSNs, machine learning determines issues such as resource allocation, security and privacy, communication cost, localization of bad nodes, and increasing network longevity [219].

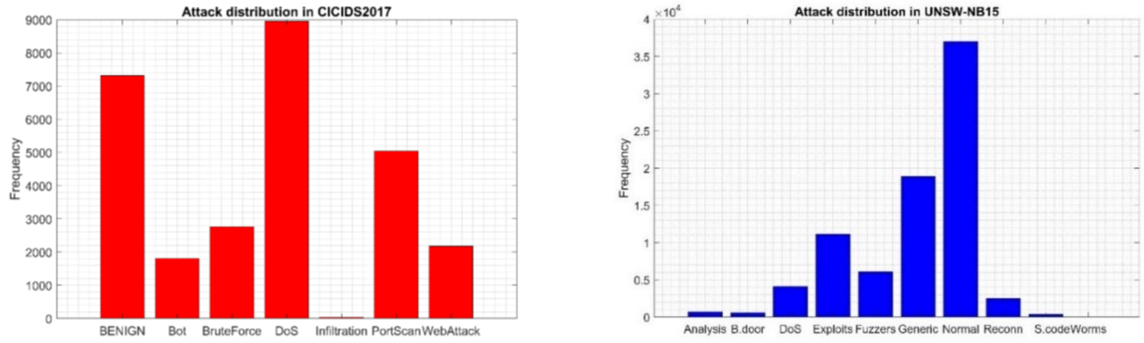
In this work, numerous learning algorithms, including a support vector machine, XGBoost, Random forest, and Ensemble stacking, are used to improve performance and evaluate the effectiveness of the proposed system using various evaluation criteria. These methods for extracting information from and discovering connections among wireless sensors are efficient and useful, and they facilitate the detection of malicious nodes in a network. As demonstrated, only nodes validated as trustworthy by the network's beacon nodes and base station are permitted inside the block, as in Figure 5. 5 in IoT-based WSNs.

Statistical algorithms that can understand complicated patterns in existing data and extrapolate those patterns to predict future data are known as machine learning models [87]. Machine learning (ML) models are developed via a training method in which the learning models analyse and interpret data logs. A wide range of digital endpoints, such as smartphones, smart appliances, sensors, and the like, generate training log sets locally in an IoT ecosystem. The quality and quantity of data used in training and testing learning models directly affect the efficacy of ML-based IDSs. Training a model locally with data sets produced by a single endpoint is called localized learning; training a model centrally with data sets gathered from multiple endpoints is called centralized learning; and training a model across multiple endpoints using local data sets without exchanging them is called federated learning.

5.4.4 Benchmark Datasets

We evaluate the proposed system's ability to recognize and categorize attacks using machine learning models on the NSL-KDD, CICIDS2017, and UNSW-NB15 benchmark datasets. This study presents a dataset for intrusion detection and network attack scenarios, which was gathered and run using tools. The dataset includes seven classes of attacks, as in Figure 5. 6(a), according to the attack sample frequency distribution. Common and typical types of attacks based on real-world facts, such as network traffic monitoring, are included in the dataset. The dataset is highly diverse, including 80 feature sets derived from CICFlowMeter assaults of varying types.

The UNSW-NB15 is a state-of-the-art intrusion detection benchmark dataset used in numerous recent studies and earlier efforts. The Australia Center for Cyber Security's cyber range laboratory IXIA PerfectStorm was used to produce the raw data traffic packets (ACCS) [151], combining "regular" network packets with "abnormal" ones. The software continuously updates the vulnerabilities and exposures of collected packets in the context of information security, simulating nine different types of assaults. With three types of category characteristics and 39 numerical features, the dataset has a total of 42 properties. The dataset contains ten different types of attacks, and their relative prevalence is shown in the accompanying diagram in Figure 5. 6(b).



(a) Distribution of assaults based on their frequency in CICIDS 2017.

(b) Distribution of attacks according to the frequency in UNSW NB15.

Figure 5. 6. Frequency distributions of attacks were plotted using the CICIDS2017 and UNSW NB15 benchmark datasets.

The dataset is split into training and testing sets to train machine learning models to make accurate predictions in classification and regression.

The NSL KDD dataset includes symbolic, numeric, and Boolean features of varied resolutions and ranges. All observable characteristics were utilized in this analysis [220]. The training and testing sets in the current NSL-KDD dataset have 148517 connection records and the same number of attributes. Using a cross-validation method, we split the data into two equal sets: 80% for training and 20% for testing. There are a total of 42 features in this data set. The last of these features (42) is a class feature. In this work, the feature engineering strategy is used for feature extraction. Nineteen features from the NSL-KDD dataset, with eighteen features extracted from the UNSW-NB15 dataset, are chosen using the suggested method.

The proposed study uses the CSV-formatted NSL-KDD intrusion dataset for testing and evaluation [135]. The NSL-KDD Dataset is utilized to evaluate the performance of the proposed system. It has multi-class data having 41 features about the characteristics of a particular type of attack on the network. This dataset consists of 34 numerical features and seven character features, and it covers "4" primary classes as indicated below: DoS, Probe, R2L, and U2R.

- With a probe, an attacker tries to learn more about the victim's network by scanning the latter's servers and other devices.
- Attacks like Denial of Service (DoS) render networks or computers unreachable to their intended users.
- With a User to Root (U2R) assault, an adversary attempts to convert a regular user into a system administrator.
- An R2L attack is one in which the attacker controls the victim's system throughout the whole network without the victim's knowledge.

The dataset has 41 features, including the protocol, duration, service, source bytes, flag, destination bytes, etc., with 80% considered for training and 20% for testing. The dataset has 38 numeric and three categorical features. The statistical distribution of the dataset is shown in Table 3. 9.

Table 3. 9. Frequency distribution attacks in the dataset.

Class of attacks	DoS	Normal	Probes	R2L	U2R	Total
Frequency distribution	45927	67343	11656	995	52	125973
Percent (%)	36.5	53.5	9.3	0.8	0	100

The CIC-IDSS2017 dataset is used as a benchmark for evaluating the performance system for detecting flooding attacks in WSNs. The dataset is available online at the Canadian Institute for cyber security research LAB. The dataset contains benign and other attacks that include network traffic data [132]. It was generated for realistic background traffic for building the dataset. The dataset was created using 25 users with different protocols. The dataset contains 485881 Instances and 31 attributes with 80% of training and 20% of testing. Five denial of service (DoS) attack classes include Slowhttptest, slowloris, Hulk, Heartbleed and GoldenEye [133]. The structure of the dataset is shown below in Table 3. 10.

Table 3. 10. Structure of the dataset with classes of attacks.

Attacks	Normal	slowloris	Slowhttptest	Hulk	GoldenEye	Heartbleed
Training	252382.6	1504.8	1276.8	127302.4	6307.2	8
Testing	63076.4	376.2	319.2	31825.6	1576.8	2
Total	315382	1881	1596	159128	7884	10

Processing is utilized to rearrange the data via normalization, missing value imputation, and aggregation for both the training and testing phases. The gaps in data are filled in using the current values' averages [134]. Data is converted from their original format into binary digits using the minimum and maximum values.

5.4.5 Dataset Pre-Processing

A raw dataset may not have undergone any preprocessing [136]. A raw dataset is incomplete, noisy, and possibly presented unfavorably. As a result, building machine learning models from scratch using a raw dataset is impossible, as shown in Figure 5. 7. The efficiency of a machine learning model can be improved by first preprocessing the raw data to eliminate extraneous information and standardizing the format. It is for this reason that this stage of a machine learning model is the most crucial. With the Data collection module, you may get traffic statistics from either a private network or the reference data set [137]. The data are sent to the preprocessing module, cleaned, and prepared for use. This information is also provided to the cluster and the trust-based safer routing module for even better data transfer. The agent that does data preprocessing makes great use of the tools. Before the preprocessing phase, data is cleaned, integrated, and modified. Cleaning and normalizing the data is the first stage in the data preprocessing phase, which aims to boost the data quality used in training and testing to develop machine learning models for prediction. Preprocessing, which includes feature extraction, feature selection, and dimensional reduction, is crucial for vectorizing [138]. Duplicate values can be removed, the mission data can be replaced, and unneeded sample structures can be eliminated. Normalization using minimum and maximum scaling values, as in equation (5.4), must be done after the dataset has been cleaned.

$$Z_{norm} = \frac{Z - \min(i)}{\max(Z) - \min(Z)} \tag{5.4}$$

Where min(z) is the minimum value and max(z) is the maximum value of the attribute Z, respectively. Z_{norm}, is a normalized feature value, and Z is an original feature value [139].

K-means cluster sampling improves the machine learning model's classification and detection performance by creating a tiny K-number of clusters from the original dataset to decrease the training complexity [41]. By eliminating unnecessary information, the K-means sampling method improves productivity, computational power, and resource use by producing highly representative small groups. Synthetic minority oversampling (SMOTE) generates superior samples to address inequalities between demographic groups. When the data has been preprocessed, feature engineering creates sensitive, high-quality features, minimizes dimensionality, and eliminates redundant features by computing the correlation features between them.

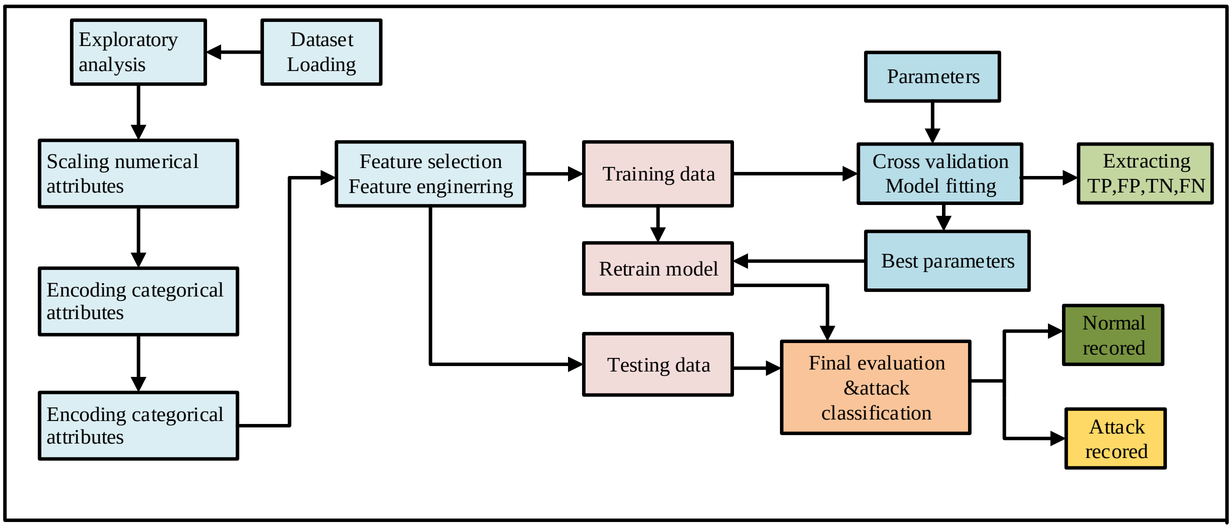

Exploratory analysis
Dataset Loading
Parameters
Scaling numerical attributes
Feature selection Feature enginerring
Training data
Cross validation Model fitting
Extracting TP,FP,TN,FN
Retrain model
Best parameters
Encoding categorical attributes
Normal recored
Testing data
Final evaluation &attack classification
Encoding categorical attributes
Attack recored

Figure 5. 7. Data pre-processing, training, and testing for model evaluation framework using benchmark datasets.

Reducing the number of attributes in the dataset and making new associations between them make the procedure less labour-intensive. Yet feature selection does not have a single, universal approach. The dataset's current condition should be considered while deciding which approach to use. Finding the best feature for discriminating between classes is the primary challenge in feature selection. Various data sets may call for distinct feature selection strategies. The feature selection method employs a plethora of different methods. Spearman's rank correlation coefficient formula is used for a recursive feature selection process, which dynamically selects features as shown in equation (5.5)

$$\rho=\frac{\sum_i (x_i-\bar{x})(y_i-\bar{y})}{\sum_i (x_i-\bar{x})^2 (y_i-\bar{y})^2} \tag{5.5}$$

Where ρ is the correlation coefficient, x_i and, y_i are the feature variables, and $\bar{x}$ and ȳ are the mean values of x and y.

5.4.6 Hybrid-Ensemble Machine Learning Techniques

Combining multiple machine learning algorithms into an ensemble makes the classification more accurate and faster. This approach involves several learning procedures using various machine-learning approaches and then combining and categorizing the results. The underlying algorithm performs two basic steps. At first, the original dataset is partitioned, and the distribution of a basic model is generated on those subsets. After doing so, the distribution is aggregated into a single model, and the results are obtained. The stacking strategy differs from standard machine learning methods because it involves a model production step. Models built from the training set are combined. You can describe the algorithm's function as follows:

- Models are created during training by employing the dataset and the training method.
- Each derived model has full annotations for all the dataset's training samples.
- The final model is built from the other models in the training dataset using the combiner method.
- After a final model has been obtained, it is used to categorize test dataset samples.
- A final prediction is made using the final model once all test dataset samples have been classified and the class predicted by the stacking algorithm of the sample is chosen.

The term ensemble technique is used to describe three distinct approaches. We're bagging, boosting, and stacking here. Data mining approaches and the capabilities of the combiner models used by each of these methods are where they diverge. As opposed to maximizing predictive power like boosting does and minimizing variance as bagging does, stacking strives to do both. The function that generates a single model uses the average weight in the bagging strategy, the weighted majority vote in the boosting approach, and Logistic regression in the stacking approach.

5.4.6.1 Gradient Boost

Extreme Gradient Boosting (XGBoost) is a fast and effective classification method for massive datasets [130]. With less memory required for training and testing the dataset for classification, the Gradient Boosting method is able to increase computational performance and produce accurate results for intrusion detection [145]. As demonstrated in equation (5.6), it is a powerful machine learning strategy for maximizing the loss function and computed features.

$$\Phi\left(X\right)=\sum_{k=1}^{K} f_k(X) \quad \mathrm{f}_k \varepsilon F \tag{5.6}$$

Where ϕ(X) is the final result of the K sequential classifier and f_{k}, is the decision tree for the K number of iterations in the Gradient descent algorithm. The method additionally employs parallel computing to categorize the required results. To optimize the process and control the overfitting factor, XGBoosting enhances the Gradient descent and regularization approach. Classifier parameters are connected using the equation (5.7) shown below:

$$\ell\left(\phi\right)_t=\sum L\left(f_{t-1}-f_t\right)+\Omega\left(f_t\right) \tag{5.7}$$

Where $\mathcal{L}(\phi)_t$, is the loss function, and $\Omega(f_t)$ is the regularization term to optimize the step size t. When retrieving scores for an attribute, the Gradient Boosting method employs feature metrics to determine how the attribute was measured.

5.4.6.2 Ensemble Learning

Ensemble machine learning algorithms use classifiers with averaged accuracy to mitigate the potential for overfitting and bias introduced by a single classifier [146]. Tree-based models used in machine learning for classification improve precision. Ensemble methods are meta-algorithms that aggregate separate machine learning prediction models for assessing stacking and variation [147]. To boost overall performance, the Ensemble method combines multiple machine learning outputs into a single, more resilient model. The fundamental ideas behind ensemble techniques are stacking, bagging, and boosting. Through stacking, we may combine and strengthen the predictive ability of multiple machine learning models.

To further improve the classification of the hybrid machine learning models for the proposed scheme on the benchmark dataset, a tree based on the Parzen estimation (PTE) is used in conjunction with hyperparameter and Bayesian optimization (BO) techniques. Hyperparameters are used in every machine learning task to fine-tune these parameters and achieve the best possible outcomes. As a result of this hyperparameter optimization (HPO), the efficiency and effectiveness of machine learning can be increased with less human input [148].

Hyperparameter optimization is also used in the black box and global optimization for more accurate function evaluation. As a result, we can explain the inner workings of Bayesian Optimization without getting too technical. In HPO for deep neural networks, Bayesian optimization (BO) is gaining popularity as a framework for global optimization with costly black-box functions. Bayesian optimization is a recursive method that uses a probabilistic surrogate model and an acquisition function to evaluate choices with the help of the Gaussian process. Random forest and tree Parzen estimators are just two tree-based approaches to deal with hyperparameters (PTE). This suggested effort combines Bayesian-based optimization (BBO) with tree Parzen estimators (TPE) to determine the optimal evaluation point for fully automated machine learning.

5.5 Performance Evaluation Metrics

The effectiveness of the proposed system is assessed using the metrics of the confusion matrix. These include accuracy, sensitivity, specificity, and training time. The new proposed approach is evaluated the evaluation metrics, including power consumption, detection rate, network lifetime, and detection accuracy of the attacks in the network. We also measure the efficacy of the suggested system concerning the following metrics: network scalability; events and communication overhead; communication range; communication failure; communication failure rate; aggregation ratio; and network load.

5.5.1 Network Lifetime

It is the operational time in which the network performs the dedicated task. It can be computed when the source node energy drains to transfer to the base station. This shows that the loss of nodes leads to the loss of network functionality [93]. Network lifetime in WSNs depends on many factors, some of which are security related while others are not. Security measures such as secure localization, routing protocols, and optimization techniques can help to increase the lifetime of WSNs, by reducing the amount of data that needs to be transmitted, as well as protecting the network from malicious attacks. Secure clustering and data aggregation techniques can also help reduce the amount of data that needs to be transmitted, thus increasing the network's lifetime. Finally, machine learning techniques can be used to identify and detect potential security threats and to adapt the network parameters to increase the overall security of the network, thus increasing the network's lifetime.

5.5.2 Accuracy and F-measure

Accuracy and F-measure are two of the most commonly used metrics for evaluating the performance of attack classification in IoT-based WSNs. Accuracy measures how accurately the system can identify and classify attacks. It is calculated by dividing the number of correctly identified attacks by the total number of attacks. F-measure measures how well the system can distinguish between different types of attacks. It is calculated by taking the harmonic mean of precision and recall, where precision is the ratio of true positives to total predicted positives, and recall is the ratio of true positives to total actual positives. Higher accuracy and F-measure scores indicate better performance in attack classification.

The reliability of a data transmission over time is measured in terms of the percentage of lost packets. How well a learning model works depends on how accurate its predictions are [193]. We conducted experiments to verify the efficacy of the suggested technique. We compared the results to those of a previously developed, secure blockchain that relied on a network of federated hybrid machine-learning models. This can be done by calculating the fraction of attacks that were correctly labelled as True Positive, the fraction of trusted nodes that were labelled as True Negative, the fraction of false positives that were incorrectly marked as True Negative, and the fraction of false negatives that were incorrectly labelled as True Positive.

5.5.3 Detection Rate

The detection rate is a positive proportion of correctly categorized normal traffic relative to the total number of samples in the collection. The true positive rate is calculated in equation (5.8). Accuracy is measured by the percentage of attacks correctly labelled compared to the total number of attack occurrences in the network. For each attack, the true positive (TP) count indicates how many times they were accurately identified as threats [194], [195], and a network's false positive (FP) rate is the percentage of attacks that were misidentified [196]. A network's "true negative" (TN) consists of all legitimate nodes properly classified as such, while a "false negative" (FN) mistakenly labels all honest nodes as malicious ones.

The Detection Accuracy is the fraction of trusted sensor nodes that correctly identify malicious nodes relative to the total number of harmful nodes in the network [6]. When compared to the

A confusion matrix creates a baseline for calculating the parameter metrics. The number of instances is tabulated as four values: true negatives (TN), true positives (TP), false positives (FP), and false negatives (FN), and expressed as in equation (5.8).

$$\text{Detection Rate(DR)} = \frac{TP}{TP+FN} \tag{5.8}$$

$$\text{Recall (RC)} = \frac{TP}{TP+FN}$$

$$\text{Precision} = \frac{TP}{FP+FN}$$

$$\text{Specificity} = \frac{TN}{FN+TP}$$

$$\text{False positive rate} = \frac{FP}{FP+TN}$$

The detection accuracy and F-Measure are also expressed as mathematically as shown below using equation (5.9):

$$\text{Accuracy(Acc.)} = \frac{TN+TP}{TN+TP+FN+FP} \tag{5.9}$$

$$\text{F-Measure} = \frac{2\times R\times P}{(R+P)}$$

To determine the False Negative Rate (FNR), which is the number of malicious nodes that the proposed model wrongly identifies as legitimate nodes, we use the following formula (5.10):

$$FNR=\frac{FN}{FN+TP} \tag{5.10}$$

In addition, this work presents new measures for evaluation, including the average localization error, coverage, localization, and detection accuracy.

5.5.4 Recall and Precision

Recall and precision are important metrics for evaluating how well an attack classification system works in IoT-based WSNs. Recall, also known as the true positive rate, measures how many of the attacks in the network were correctly identified by the system. Precision, also known as the positive predictive value, measures how many attacks the system identified were present in the network. A high recall rate indicates that the system can correctly classify most of the attacks in the network. In contrast, a high precision rate indicates the system can identify most attacks it detects correctly. Thus, a high recall and precision rate is necessary for an efficient attack classification system in IoT-based WSNs.

5.5.5 Localization Error

The localization error quantifies the distance between the estimated and actual locations [197]. The radio range of sensor nodes normalizes errors in localization to provide a consistent metric for comparison. Average localization error (ALE), Considering all of the nodes in the study area, we can calculate an average localization accuracy to characterize how well we can place them [198]. Several metrics, including the average detection rate, accuracy, precision, and recall, measure the suggested system's efficiency. The ALE can be computed by adding up the LE of each unknown node and then dividing that sum by the total number of unknown nodes [170]. The LE measures how far away a node's predicted position is from its actual physical location. The average localization error (ALE) and average localization accuracy (ALA) are used as evaluation metrics. The average error localization, shortened as ALE [3], is computed in equation (5.11), respectively. The ALE is calculated by adding the LE of each unknown node and dividing it by the total number of unknown nodes. Specifically, the LE is defined as the discrepancy between the predicted and observed locations of black boxes.

$$\text{Localization Error(LE)} = \sqrt{\left(\mathrm{u}_i' - \mathrm{u}_i\right)^2 + \left(\mathrm{v}_i' - \mathrm{v}_i\right)^2} \tag{5.11}$$

$$\text{Average Localization Error(ALE)} = \sum_{i=1}^{n} \frac{\sqrt{\left(\mathrm{u}_i' - \mathrm{u}_i\right)^2 + \left(\mathrm{v}_i' - \mathrm{v}_i\right)^2}}{nR}$$

$$\text{Average Localization Accuracy(ALA)} = \left(1 - \left(\sum_{i=1}^{n} \frac{\sqrt{\left(\mathrm{u}_i' - \mathrm{u}_i\right)^2 + \left(\mathrm{v}_i' - \mathrm{v}_i\right)^2}}{nR}\right)\right) \times 100\%$$

Where $\left(\mathrm{u}_i', \mathrm{v}_i'\right)$ are the actual coordinates of the anonymous node i and $\left(\mathrm{u}_i, \mathrm{v}_i\right)$ are the computed coordinates of n unknown nodes, and R is the radius of communication in the network.

5.5.6 Intimacy and Honesty

Intimacy and honesty are two important security aspects in IoT-WSNs (Internet of Things-Wireless Sensor Networks). Intimacy refers to the need to establish trust between nodes in the network. This ensures that only authorized entities can access the network and that communications between nodes are encrypted and secure. Honesty means ensuring the data being transmitted and received is accurate and valid. This includes verifying the integrity of the data and ensuring that malicious actors cannot interfere with communications. These two security measures ensure that IoT-WSNs are secure, reliable, and trustworthy. It is computed by using the following formula as in (5.12):

$$\mathrm{ln} = \frac{\mathrm{t}_i}{\mathrm{t}_i + \mathrm{t}_a} \tag{5.12}$$

Where t_a, is the time consumed for a node to communicate with the anchor node and t_i, is the collaboration time of a particular node with beacon node i.

Another metric that is used to calculate behavioural trust is honesty. The equation (5.13) used to calculate it is presented below.

$$h_o = \frac{p_s}{p_n} \tag{5.13}$$

There are P_n interactions between source and destination nodes, and $P_s$interactions between source and destination nodes that result in successful outcomes. The sincerity of a statement reveals the total number of interactions between nodes, both successful and unsuccessful. The interactions factor in round-trip latency as a criterion for grouping communications between nodes. The term "end-to-end latency" describes the average time it takes for valid and faulty data packets to travel from source nodes to sinks [221]. A data packet's latency is the time it takes to travel from its source to its destination, where it will remain indefinitely.

5.5.7 Friend Nodes Interaction and Trust Degree

The rate of interactions is measured by how often anchor nodes communicate with one another. An important statistic for computing behavioural trust is the ratio of the overall number of one-hope nodes in every beacon node to the entire number of nodes excluding this particular beacon node. This path connects the beacon node to the rest of the network as quickly as possible. The range of an anchor node is defined as the set of all other nodes within some specified distance from the anchor node. Every beacon node is assigned a trust value used to select the safest and most dependable ones [19]. Several nodes in the network are unsure of their exact location, as determined by the trust value computation. Although they can't find their way through the network without the assistance of beacon nodes, these nodes are expected to be the most trustworthy and reliable overall. Triangulation and distance vector routing may determine the most trustworthy beacon nodes. The sum of the trust ratings for behaviour, responses, and reliable information. Trust in behaviour is evaluated along four dimensions:

Behavioural trust in beacons is lost when malicious nodes are within range of any anchor node. The overall trust in beacon nodes is harmed, leading to the wrong choice of anchor nodes and cluster head nodes for the precise location of malicious nodes in WSNs. Where malicious nodes can be located in WSNs depends on several factors, including the number of neighbour nodes, distance error, security, mobility, and localization approaches [222].

Every beacon node's feedback trust is computed after the established behavioural trust. The other nodes can't find the beacon or unknown node unless the beacon or unknown node shares its coordinates. This node offers feedback on the sender beacon after confirming that the sender beacon provided accurate coordinates. The data-based trust of each node in the network of beacons is then determined by comparing the actual distance between beacons to the predicted distance. Euclidean theorem, provided by equation (5.14), can be used to pin down precise distances.

$$D_{ac}=\sqrt{\left(u_i - u_j\right)^2+\left(v_i - v_j\right)^2} \tag{5.14}$$

Where D_{ac} is the actual distance u_i and v_i denote the positions of the sender anchor node and u_j and v_j, are the positions of the destination anchor node.

5.6 Simulation and Testing

The primary experimental programming languages we used were Python 3 and MATLAB [79]. To evaluate the efficiency of the proposed technique on the dataset, we apply IBM SPSS and the WEKA Java toolboxes for data processing and analysis. The proposed method has been implemented using well-known libraries like NumPy, simplifying the manipulation of multi-dimensional arrays and matrices. Attacks can be analyzed and categorized with the help of matrices and arrays like these. Pandas provide easy access to potent analysis tools and the ability to manipulate data structures readily. Since their introduction, frameworks like TensorFlow and Keras have spread throughout the deep learning and machine learning communities. The Scikit-learn library can be used to create ML methods in both supervised and unsupervised settings. SMOTE was developed to select a larger sample size from under-represented demographics methodically. The models' networks were planned and simulated using MATLAB R2021a on a computer with an Intel(R) Xeon(R) Silver 4214 CPU at 2.20GHz 2.19 GHz (2 processors), with 128 GB of installed RAM (128 GB of usable RAM), running Windows 10 64-bit on an x64-based processor. Python is used for scripting, web development, and processing and analyzing data.

Table 5. 1 displays the deployment simulation setup's parameters and runs a full security assessment on the simulated network. Similarly, the Solidity programming language is used to develop smart contracts and consensus procedures. To reach a consensus across a network, both Proof-of-Work and Proof-of-Audit are implemented. The performance of the created algorithms is measured in a variety of ways. They include gas consumption, transaction delay, geolocation accuracy, and classification precision Table 5. 1 provides the simulation parameters used to test our proposed models. Ten separate IoT-WSN zones have been implemented in this proposed blockchain-based localization for detecting malicious attacks and varied simulation settings for the proposed models. In these setups, nodes are chosen beforehand and are responsible for approving transactions and adding blocks to the blockchain via the Proof of Authority consensus mechanism.

Table 5. 1. Simulation setup for the proposed network model based on blockchain technique.

Parameter	values	Parameter	values
Software	MATLAB	Total sink nodes	10
Number of sensors	1600	Number of clusters	53
Deployment Area	20000 ×20000 m^2	Mobility	Random
Protocol type	Clustering and routing	Attacks	Routing
Total unknown nodes	1480	Sink position	500, 2000
Total beacon nodes	110	Transmission radius	250
Total edge servers	10	Data size	4000 Kb

For IoT-based WSNs, the suggested localization method uses blockchain to identify malicious assaults, and the models are distributed among ten distinct IoT-based regions. The region uses PoW and PoA as consensus mechanisms, validating transactions and adding blocks for malicious node identification and service delivery. Trusted nodes validate transactions and add blocks that can be used to track down attackers and stop the spread of attacks. This blockchain-based method allows locally-targeted routing assaults, which are detrimental to the availability and throughput of a network. The suggested methodology is new because it utilizes the intelligent communication of the wireless sensor nodes within a particular region to assess scalability and chain of trust. Since the system's security performance varies depending on the area, distributed denial-of-service attacks cannot compromise numerous faiths simultaneously.

5.7 Result and Discussion

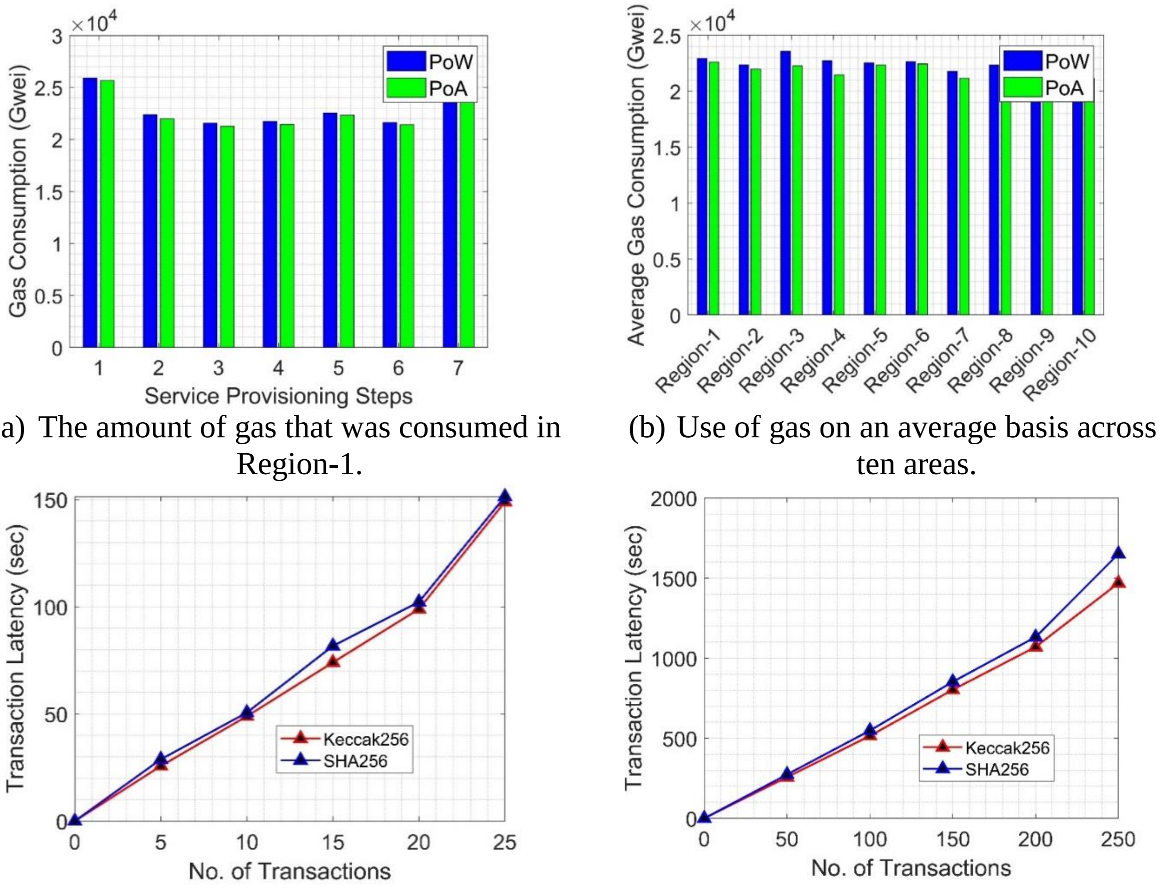
In this section, the experimental results based on localization techniques and blockchain techniques are discussed in detail. The results of secure localization in WSNs are positive. Implementing secure localization ensures that the node’s location is accurately determined using the DV-hop algorithm, RSS, DE, and their combinations for secure communications and detection of malicious nodes. This allows for a high level of security against various types of attacks. Furthermore, secure localization allows the network to track the nodes' movements accurately. Secure localization allows the network to track the nodes and their activities accurately. Furthermore, secure localization can prevent malicious nodes from entering the network and stealing data.

5.7.1 Discussion on Blockchain Techniques

Furthermore, for ten autonomous and non-repudiation zones in IoT-based wireless sensor networks, the results demonstrate the efficacy of the suggested approach for malicious node detection and localization utilizing blockchain technique. The set of the network's nodes, as shown in Figure 5. 8 (a) and Figure 5. 8(b) depend on a proof-of-authority (PoA). The future IoT-based wireless sensor networks will rely on a proof-of-authority (PoA) consensus mechanism to verify transactions and append new blocks to the chain. In most consortium blockchains, PoA is used as the consensus mechanism. A group of validators is selected via Proof of Authority (PoA) to add new blocks to the blockchain. The system uses reputation to select validators. PoA requires far fewer computational resources than PoW because it allows validators to be preselected. Mining is not a part of the PoA consensus mechanism at all, in contrast to PoW. Public blockchains like Bitcoin and Ethereum use a consensus method called proof-of-work (PoW). Proof-of-Work selects a miner through a cryptographic problem promulgated across the network (PoW). The winning node is the one that most efficiently and precisely solves the problem. One miner is selected to validate all transactions and generate the block hash. The hash is shared with every node on the network to facilitate consensus. Figure 5. 8 (a) shows the variance in gas consumption in region-1 IoT-WSNs between power levels and areas where services are provided. This shows the variance in gas consumption for region-1 IoT-WSNs based on power and location. Figure 5. 8 (a) shows how Region-1 of an IoT-based wireless sensor network uses gas differently for safe service provisioning between PoW and PoA. For the gas, you'll need a negligible amount of cryptocurrency. The person whose account is being used in the Ethereum blockchain transaction has their funds debited [223]. The user's cryptocurrency balance gets reduced. Figure 5. 8 (a) displays the events' gas usage in terms of gas units during the secure service provisioning procedure.

For example, Figure 5. 8 (b) examines the standard gas usage of IoT-WSNs across all ten areas of PoW and PoA. It's clear from these results that PoW consumes more energy than PoA. This is because, under Proof-of-Work (PoW), the miner nodes responsible for validating transactions are selected through a rigorous mathematical process. Yet, in PoA, miners are chosen not based on their skills but on how much money they are willing to risk. For each block added to a blockchain, a group of miners is selected to verify its transactions.

Similarly, Figure 5. 8 (b) examines the gas used by PoW and PoA wireless sensor networks based on the Internet of Things across all ten regions. As can be seen from the data, PoW has a higher relative gas consumption than PoA. This is because the miner nodes responsible for validating transactions are selected in Proof-of-Work (PoW) using a complex mathematical process. In contrast, in PoA, miners are chosen not based on their skills but on the size of their bets. For the blockchain to continue functioning, a group of miners is selected to verify transactions and add new blocks. The PoA technique compromises the decentralized nature of the blockchain because all transactions must rely on a small set of miners. This is because Proof-of-Authority (PoA) increases network centralization in blockchains.



(a) The amount of gas that was consumed in Region-1.

(b) Use of gas on an average basis across ten areas.

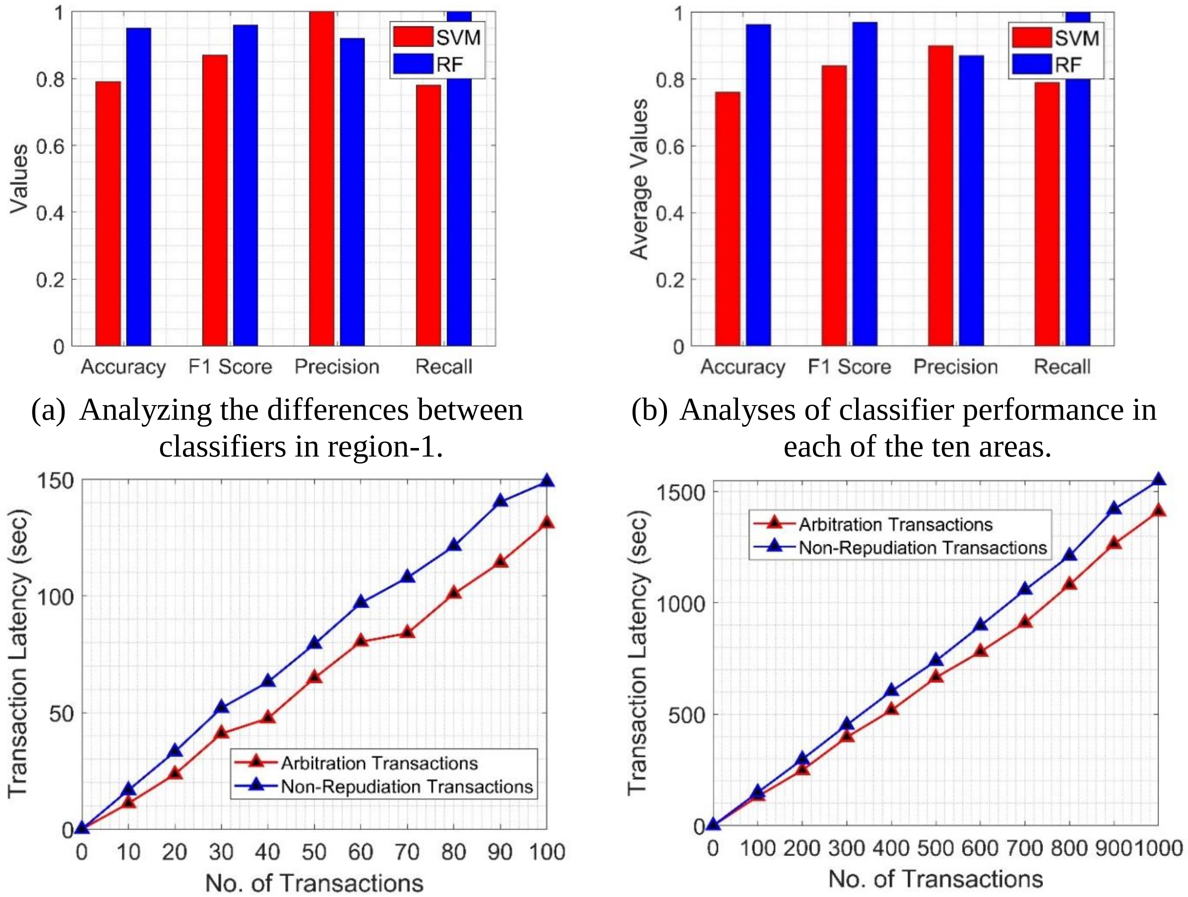
(c) The transaction latencies within region-1.

(d) Comparison of transaction latencies among ten areas.

Figure 5. 8. The amount of gas each region uses and how long transactions take to complete in IoT-based wireless sensor networks.

IoT-based wireless sensor networks in region-1 and 10 IoT-based wireless sensor networks' transaction delays using the Keccak-256 and SHA-256 hashing algorithms are compared and contrasted as in Figure 5. 8 (c) and (d). The Keccak256 hashing algorithm extends the Secure Hashing Technique (SHA3). The Keccak256 algorithm is pre-implemented in the Solidity programming language. All data can be hashed into a fixed-length hexadecimal value using Keccak256. This hash cannot recover the original data because the hashing process is irreversible. Keccak256 has a lower cost profile than other hashing algorithms like SHA256 and RIPEMD160. Because of this, we've decided to use it. In region-1 of the network, sensors, as in Figure 5. 8 (c), perform 25 transactions per second, while all Figure 5. 8 (d) sensors perform 250 transactions per second. Figure 5. 8 (c) and (d) illustrates a system with a single area of IoT sensors and one with ten. The figures exhibit the same behaviour as the SHA-256 and Keccak-256 hashing algorithms. The numbers show that executing SHA-256 takes nearly twice as long as Keccak-256. Keccak-256 does not have to deal with an unlimited input space makes it superior to SHA-256 for hashing purposes.

Scalable and extensive networks benefit significantly from a non-repudiation approach that makes use of support vector machines, XGBoost classifiers, random forests, and ensembles of classifiers. Using federated machine learning methods, the model also provides efficient procedures for identifying and pinpointing rogue nodes in the network model. Data points can be divided into two groups using linear separability in the SVM hyperplane. The most trustworthy candidate is the category with the highest honesty value and the shortest total time to completion. Malicious nodes, on the other hand, cause a lot of extra wait time and have a low integrity rating. In IoT-based wireless sensor networks, the recall, precision, accuracy, and F1 score of RF and SVM are displayed in Figure 5. 9 (a) and (b). When applied to IoT wireless sensor networks, RFID improves the efficiency of detecting malicious nodes. Figure 5. 9 (a) shows that SVM has an accuracy of 89%, but RF has an accuracy of 99%. According to these results, the average accuracy of the support vector machine (SVM) is 86%, whereas the average accuracy of the RF is 99.3% overall in ten regions of interest. The results show that, when comparing RF and SVM, RF performs better at identifying malicious nodes in networks from different areas in terms of both integrity and latency. The superior accuracy of RF can also be attributed to ensemble learning, which consists of several decision trees.



(a) Analyzing the differences between classifiers in region-1.

(b) Analyses of classifier performance in each of the ten areas.

(c) The transaction latencies within region-1.

(d) Comparison of transaction latencies among ten areas

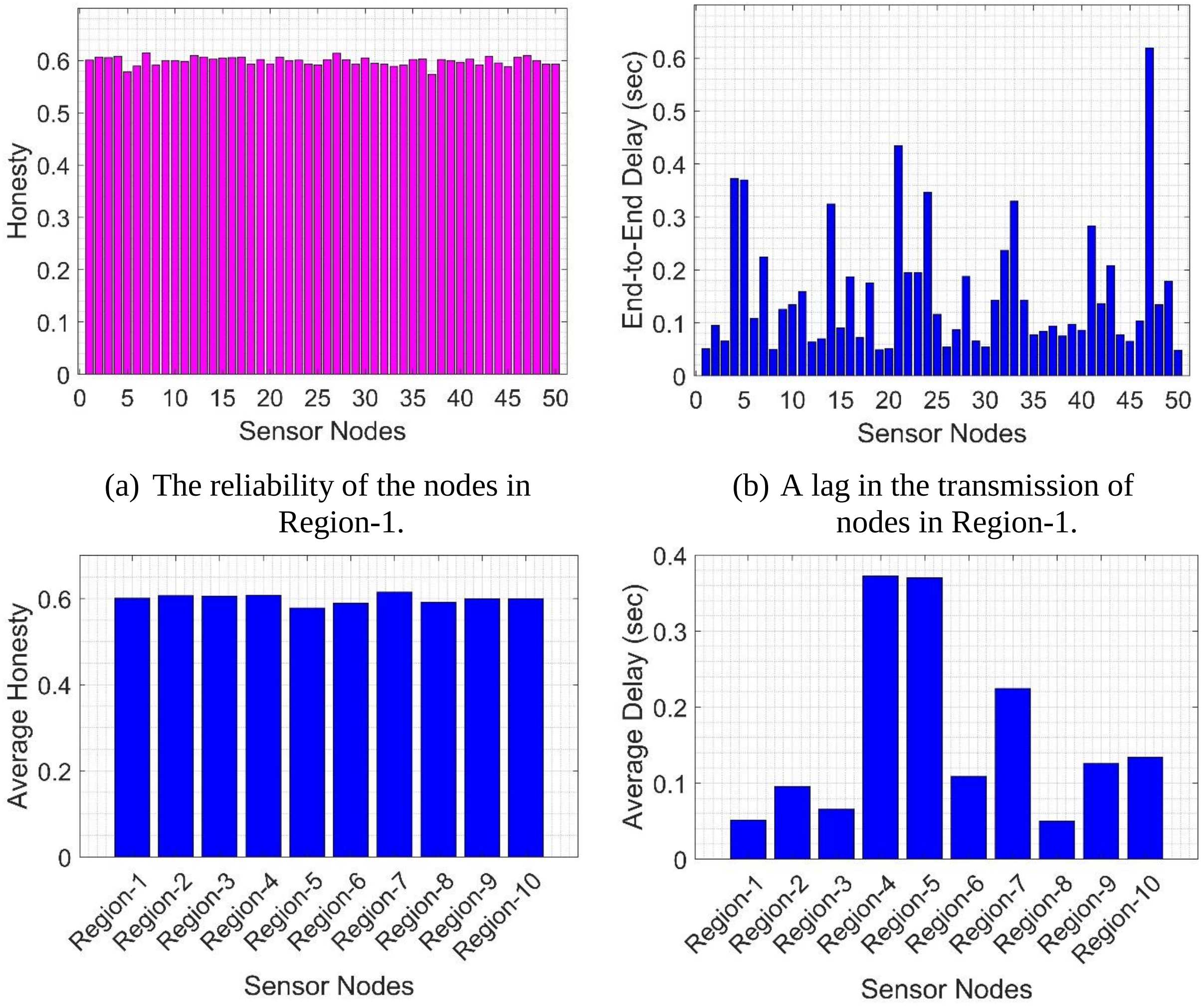
Figure 5. 9. Performance compression of classifiers and transaction delay utilizing blockchain approach for detecting and localizing rogue nodes in IoT-based WSNs.

As a result, many trees are generated to train and evaluate the same solution to the problem of detecting malicious nodes. Figure 5. 9 (a) shows that the F1 score of SVM in a single region is 87.95%, while RF has a 96% F1 score. As depicted in Figure 5. 9 (b), Average F1 scores for SVM and RF in all ten IoT-based WSN locations are 84% and 97%, respectively. This makes it less likely that RF signals from malicious nodes in sensor networks would be mistaken for those from legitimate nodes, whether the networks are localized in a single location or cover a large geographical area.

In addition, as shown in Figure 5. 9 (a) and (b), RF is superior at predicting hazardous nodes, so its recall score is higher than SVM's. Also, this demonstrates that SVM has higher accuracy than RF. Because of its support vector, SVM can accurately identify the positive observations. All ten regions of IoT-WSN clusters were demonstrated to benefit from our proposed non-repudiation approach in Figure 5. 9 (c) and Figure 5. 9 (d). The numbers show a wide range of disagreements. The latency of a transaction refers to how long it takes for a network to complete a transaction. The lag time clients and service providers experience during arbitration transactions is also shown. It turns out that both the service provider and the customer are the bad guys in these exchanges. Following the schematics in Figure 5. 9, Our proposed system can handle several transactions rapidly. So, making an objective comparison without disclosing any information about the characteristics listed or the service provided is possible. This illustrates that the arbitration transactions use negligible throughput and bandwidth. Non-repudiated transactions also show a low latency, as shown by the results. This demonstrates that our proposed non-repudiation technique is efficient regarding service quality and security while protecting many users across many IoT-WSNs.

Nodes from region-1 IoT-WSNs are shown to be honest in Figure 5. 10 (a). Random selection yields a sample of fifty nodes from the IoT-WSNs in region-1. The graph depicts a trust hierarchy, with each node representing a different degree of reliability. Because of this, the quantity of productive interactions at each node is different. Each of the ten IoT-WSNs' Integrity can be determined using the same formula, as seen in Figure 5. 10 (c). Nodes with a high honesty rating are more likely to be legitimate. This suggests the node is highly connected to other sensor nodes in WSNs through various channels. Integrity is a cornerstone evaluation criterion for validating links in our networks. Transmission, propagation, processing, and queuing delays can all be calculated to verify a node's validity. Any nodes whose end-to-end latency exceeds a specific threshold value are malicious nodes. When contacted by other nodes, nodes take too long to react. Also, they cannot process the data packet due to their restricted processing capabilities, which leads to a denial of service.

There is a high end-to-end delay with this type of node that broadcasts this information (latency). Figure 5. 10 shows the end-to-end latency of region-1 IoT-WSNs and the average end-to-end delay of all ten regions' IoT-WSNs. Figure 5. 10(b) and Figure 5. 10(d). The figure depicts the random selection of fifty ordinary nodes from each location. Figure 5. 10 (b) demonstrates that the round-trip time between nodes varies. This is because nodes' response times to queries vary. Nodes with an end-to-end delay greater than a certain threshold are considered malicious in this scenario. Another intriguing result from Figure 5. 10 is that the honesty and end-to-end delay average values for different IoT-WSNs region networks span a wide range, illustrating the varying degrees to which these networks succeed in ensuring the safety of their users.



(a) The reliability of the nodes in Region-1.

(b) A lag in the transmission of nodes in Region-1.

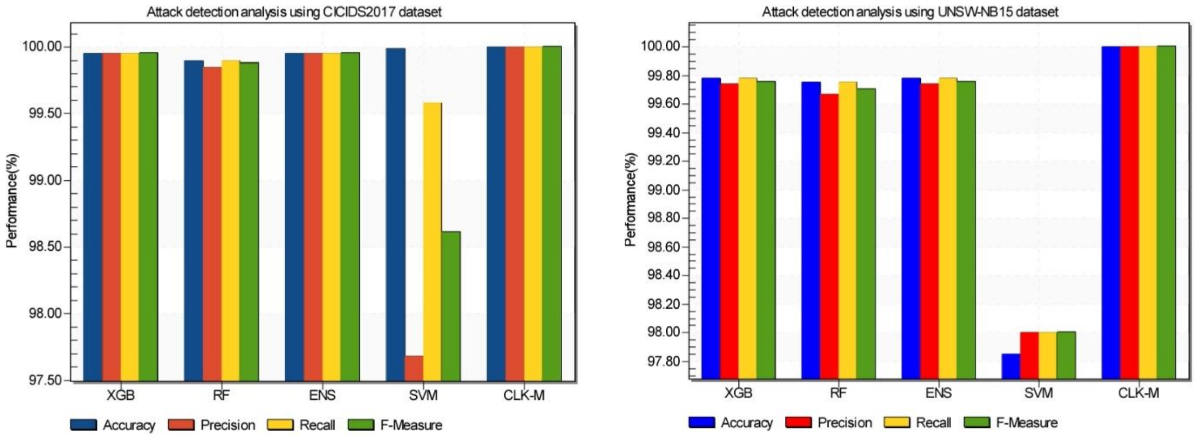
(c) Honesty rating of nodes on average over ten regions

(d) The average amount of time that nodes in ten Regions spend waiting

Figure 5. 10. Honesty of wireless sensor nodes and end-to-end latency of nodes in the Internet of Things and Wireless Sensor Networks (IoT-WSNs).

5.7.2 Discussion Based on Federated Learning

With the use of machine learning models and a benchmark dataset including a class of attacks in WSNs, the effectiveness of the proposed solution is measured. The efficacy of the proposed routing attack localization and detection in wireless sensor networks is assessed using the same benchmark dataset by the hybrid optimized machine learning. Figure 5.11 (a) highlights how various machine-learning methods stack up against one another. Cluster labelling (CL) k means binary classification methods are used to boost the suggested system's performance further. Cluster labelling (CL) with k-means binary classification can improve recommendation and information retrieval systems. Labelling k-means clusters is part of CL with binary classification. This method gives clusters meaningful labels that can increase system performance. Cluster labelling with k-means binary classification can increase recommendation relevance and personalization. However, this approach's efficacy depends on several parameters: feature quality, clustering technique, binary classifier performance, and labelled data availability.



(a) Hybrid machine learning modes using CICIDS2017.

(b) Hybrid machine learning modes using UNSW_NB15.

Figure 5.11. Several performance metrics were used to evaluate numerous machine learning models using CICIDS2017 and UNSW NB15 benchmark datasets.

Furthermore, a UNSW NB15 benchmark dataset is used to assess the proposed scheme's efficacy after being subjected to several machine-learning models, as in Figure 5.11 (b)

The hybrid cluster labelling K-means binary classification approach achieves better classification accuracy on the benchmark dataset. As shown in Figure 5.11, the suggested system's performance is sufficient for attacks on IoT-based wireless sensor networks, as measured by the benchmark datasets. Validation and confirmation can be attained by comparing the simulation and experimental results to previous efforts in the field.

As shown in Figure 5. 12, Z. Abubaker et al. [19] suggested a blockchain-based method for detecting and localising rogue nodes in IoT-based WSNs by employing federated machine learning with support vector machines and radio frequency. The illustration demonstrates that the suggested method, which utilizes a hybrid federated machine learning methodology, effectively detects and localizes malicious nodes in IoT-WSNs. S. Salim et al. [224] provide a differentially private blockchain-based explainable FL (DP-BFL) architecture illustrated in Figure 5. 12. This design was developed with the dynamic power of SM 3.0 networks and MNSIT datasets as a benchmark assessment method. Due to the existence of this platform, data may now be contributed to a globally secure model by any device that has access to the internet. The article by A. Rehman and colleagues [225] gives a comprehensive analysis of the findings from their investigation into the relationship between blockchain technology and federated learning in the healthcare industry. The objective of this study is to develop a reliable monitoring system for healthcare. They were utilizing a blockchain-enabled, federated-learning-based Intrusion Detection System (FL-IDS) to detect any malicious activity within a healthcare network, allow physicians to keep tabs on patients via sensors, and take preventative measures regularly by foreseeing diseases, as shown in Figure 5. 12. The findings of the proposed system demonstrate that the approach is suitable for use in healthcare monitoring.

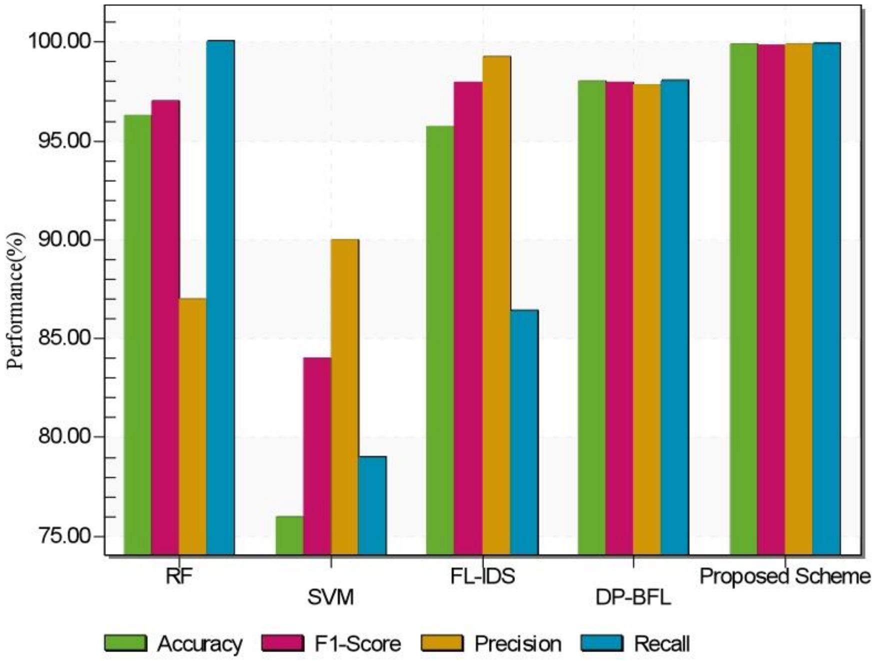
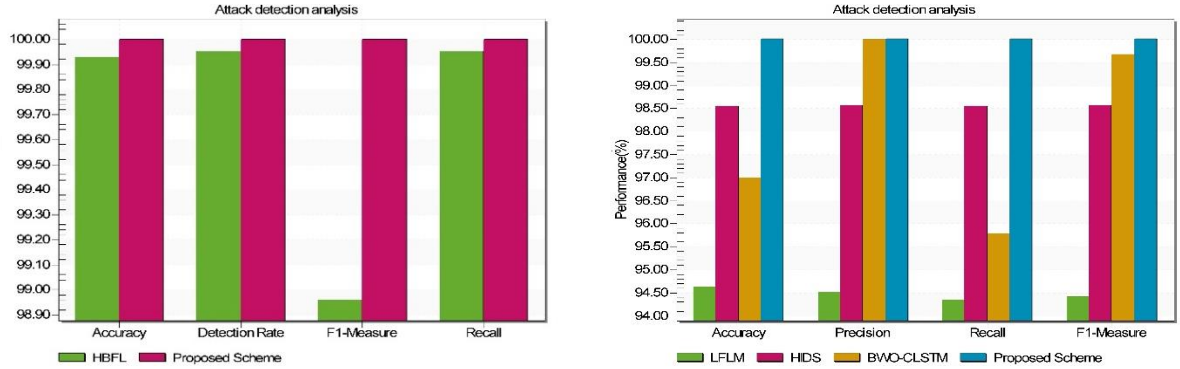


Figure 5. 12. Analysis of the effectiveness of the suggested approach in detecting malicious nodes in IoT-WSNs using a benchmark dataset and comparing the results to those obtained using alternative evaluation metrics.

The findings indicate that the proposed scheme is superior to the one S. Awan and colleagues developed. A trust evaluation process based on blockchain was provided in reference [74] to record the identities of sensor nodes (SN) and aggregator nodes (AN). The proposed method achieves a detection accuracy of 95%, bringing the number of malicious nodes down to 5-30.

The existing method achieves a detection accuracy of 75%, taking the number of malicious nodes to 20-80. The current process utilized private and public blockchains to detect malicious nodes, resulting in a detection accuracy of 75%. S. Otoum et al. [76] proposed a trust evaluation and security paradigm by combining blockchain and federated learning. An average detection rate accuracy of 93% and 96%, respectively, was reached using this approach. T. H. Kim et al. [16] used a secure blockchain based on trust management and evaluation. They achieved an average localization detection accuracy of 96.5% and an average localization error of 6.97, respectively. O. Friha et al. [79] constructed DNN, CNN, and RNN learning approaches. They achieved an average detection accuracy of 98.63%, 99.71%, and 99.05% for FELIDS by utilizing a benchmark dataset for testing the detection performance of the system. These results are presented as percentages. Collaborative security system-based federated learning for the Internet of Things was provided in S. Kim et al. [138].'s research work. They used the NSL-KDD benchmark dataset, which included various classes of attacks for training and testing, and achieved a training accuracy of 98.47%. This would imply that the proposed approach is improved for detecting and localizing attacks in IoT-based WSNs due to the findings. The method based on blockchain technology developed for attack detection and localization enhances the security level compared to A. Ahmed et al. [219], it lowers the amount of energy consumed and effectively aggregates data by using the trust values of the beacon nodes. This increases network lifetime. Z. Mahmood and V. Jusas [226] presented blockchain-enabled federated learning for data security and privacy in IoT-WSNs and decentralized nodes and sent worldwide data sharing. They obtained an average detection accuracy of 95% with their system. In addition, the system aggregates the data for secure transactions by moving it from the local model to the global one. This demonstrates that the strategy that was proposed achieves higher classification accuracy.

Figure 5. 13 (a) depicts a proposed work by M. Sarhan et al. [87] for a hierarchical blockchain-based federated learning (HBFL) to create a secure collaborative intrusion detection system in IoT-based wireless sensor networks (a). When tested on the UNSW-NB15 benchmark dataset, which includes multiple types of assaults, the scheme's average detection accuracy was 99.71%. T. Hassan et al. [227] presented using a nonlinear support vector machine (SVM) classification model; the authors propose a self-learning and adaptive protocol for IoT-based WSNs to automatically transmit multi-user data by efficiently using channel spectral properties. In a collaborative yet private manner, N. Nasser et al. [228] presented a lightweight federated learning model (LFLM) to learn medical symptoms with high accuracy from data collected by individuals using ambient IoT based-sensors and wearable devices for the covid-19 response, with evaluation metrics a shown in Figure 5. 13 (b). Using machine learning, S. Roy et al. [229] describe a two-layer hierarchical intrusion detection system (HIDS) that can identify intrusions in IoT networks while remaining compliant with the IoT's resource limitations. To effectively deal with these issues in IoT-based WSNs, P. R. Kanna and P. Santhi [38] proposed using MapReduce to construct black widow-optimized convolutional-long short-term memory (BWO-CONV-LSTM) networks for a hybrid IDS model.



(a) Performance comparison with M. Sarhan et al. [87].

(b) Performance comparison with various schemes.

Figure 5. 13. Analysis of the suggested scheme's performance in relation to state of the art utilizing several evaluative indicators and a benchmark dataset.

According to K. P. S. Kumar et al. [193], a federated machine learning-based intrusion detection system was proposed to identify and locate malicious nodes. Using KDD-99 in IoT-based WSNs, they improved detection accuracy to 92.7% on average. S. S. Mohar et al. [230] and A. Kumar [231] accomplished an average localization error of 0.43 with four beacon nodes utilizing a hybrid scalable, secure collaborative localization strategy for WSNs, which is compared further with the suggested approach. M. B. E. Sajid et al. [241] presented GA-SVM and GA-DT, two Genetic Algorithm-based models for spotting bad actors. The Dijkstra algorithm finds the quickest path through a network once a bad actor node has been located. The success rates of GA-DT and GA-SVM are also very high, at 96% and 98%, respectively. The GA-DT and the GA-SVM are highly accurate at 94% and 96%, respectively. Using 114 samples, T. J. Nagalakshmi et al. [232] developed a machine-learning model for detecting and localising black hole attacks in WSNs, with a resulting localization accuracy of 98.01%. This reveals that the proposed method's performance is adequate for detecting and classifying attacks in IoT-based WSNs.

5.8 Conclusion and Remarks

In order to solve the security issues, we proposed to use a security technique based on blockchain technology to locate and detect malicious nodes present in hierarchically decentralized IoT-WSNs. Utilizing XGBoost and CLK-Means machine learning federated classifiers for multiclass and binary classification approaches, blockchain technology can identify and localize rogue nodes when applied to IoT-based wireless sensor networks. This enables the technique to identify and localize rogue nodes successfully. Cascade encryption and feature assessment were utilized in this strategy to guarantee the delivery of secure services at increased network speeds. The proposed system effectively detects and localizes malicious nodes, with average detection accuracies of 99.95% and 100% for XGBoost and CLK-Means, respectively, using the multiclass and binary classification approach with the CICIDS2017 benchmark dataset. These results were obtained through simulation and classification. Using a hybrid strategy powered by machine learning to detect and localize attacks on IoT-WSNs is a novel innovation. According to the findings, random forest performs significantly better than alternative methods when classifying complicated network traffic data and locating rogue nodes. The proposed model is evaluated using several measures, such as node honesty, end-to-end delay, and transaction latency. These metrics all speak to how well the model delivers services in a timely and secure manner. Because sensor nodes deployed in unattended systems are inherently susceptible to a wide variety of routing attacks, there is an urgent need for research into the most effective ways to determine the specific location of malicious nodes.

Our long-term objective is to research and develop state-of-the-art blockchain-based secure IoT-WSNs using hybrid federated and machine-learning methodologies for large-scale rollouts of secure and intelligent IoT-WSNs. This will be accomplished by studying and developing these systems. As part of our investigation into blockchain technology, we will investigate complicated hybrid access control methods. These techniques may be applied to IoT-WSNs to find and recognize malicious nodes. On top of that, we will construct state-of-the-art multiclass and binary classification assessment metrics by using a wide range of benchmark datasets in conjunction with hybrid routing protocols.

Blockchain technology offers an immutable and distributed ledger for safeguarding and exchanging sensitive information. Secure localization is made possible through blockchain in IoT-WSNs, which verifies the accuracy and integrity of location information. Blockchain's decentralized design eliminates the possibility of centralized data modification or hacking by rendering the network inaccessible to any one organization. Blockchain's immutability and transparency ensure accurate and trustworthy localization information. The immutability of blockchain records means that location data stored there cannot be changed without the agreement of all network nodes. This enables Internet of Things devices to rely on reliable and immutable location data, which is essential for a wide range of uses, including asset tracking, supply chain management, and smart city infrastructure. Blockchain is very secure and difficult to compromise because of its decentralized structure and consensus mechanism. Using blockchain-based secure localization can reduce risks in IoT-WSNs, where devices are susceptible to a variety of security attacks. Despite compromised devices or malicious nodes, the decentralized consensus method guarantees that most nodes confirm and maintain the integrity of location data.

Chapter 6

6 SECURE CLUSTERING, DATA AGGREGATION, AND ROUTING STRATEGIES

Secure clustering, data aggregation, and routing in IoT-based wireless sensor networks assure network security, dependability, and scalability. These methods efficiently transfer, combine, and route data via the network, reducing data breach risk. Sensor nodes can cluster to save energy and provide larger data sets. Multi-hop and decentralized routing techniques safeguard data transmission and storage between nodes. These solutions increase network performance and security.

6.1 Introduction

Wireless Sensor Networks are vastly dispersed networks formed by many lightweight, small sensors in which each sensor is arranged and equipped for detecting and monitoring physical phenomena. WSNs are multi-hop, self-organizing networks of sensor nodes [233]. WSN nodes work together to store and process sensing data from the monitoring area. Sensor networks are utilized in defence, transportation, medicine, health, and environmental monitoring. WSNs are small and low-cost sensor nodes with power and energy applied for environmental and health, home automation and appliance management, and military examinations. The three main components of a WSN are the base station, cluster heads, and sensor nodes. The network model consists of several clusters with a monitoring node at the center acting as the node manager and responsible for the aggregation and clustering node [91], [92], [234]. The sensor nodes utilize wireless radio channels for sensing, data processing, and communication. Sensor nodes are also used for event monitoring and report generation to the aggregation clustering node. The data is aggregated and processed to the sink node via the cluster head for classification using machine learning classifiers. Depending on the location of the sink node, the data can be sent to the base station via the intermediate and cluster head. Sensor nodes have limited resources and minimize data transmission to improve resource utilization and network lifetime. This efficient data aggregation technique is essential for minimizing communication overhead in data transmission and security since security is a crucial issue in WSNs. They are deployed in unattended environments for transmitting sensitive information. Since one of the primary purposes that the sensor nodes are intended to serve is the collection of data from the environment [235], the data generated from sensors are typically excessive in quantity and frequently redundant. As a result, we need a method for combining the data that has been sensed to provide some useful information, and we can accomplish this goal by aggregating the data. It is believed to be the way of aggregating the data from various sensors to eliminate redundant communication, estimate the desired answer about the phenomenon being felt, and then provide the fused and secured information to the base station.

WSNs are exposed to several security threats due to the sensor networks' nature and operational resource constraints [236]. Sensor nodes are susceptible to multiple attacks due to the broadcasting and hostile nature of the environment. Some of the attacks in WSNs include wormhole, Sybil selective forwarding, jamming, exhaustion, sinkhole, clone, routing, track, and collision attacks. Sybil attacks are the most harmful attack for WSNs creating multiple identities, as shown in Figure 6.1. Sybil attacks degrade the performance of attack detection, resource allocation, routing, system synchronization, and data aggregation of wireless sensor networks. Sybil attacks present multiple identities that confuse the network that directly and indirectly communicates using fabricated and stolen identities. This attack targets the legitimate node with multiple fabricated identities in the network. At the same time, the wormhole attack creates tunnels between two malicious nodes.

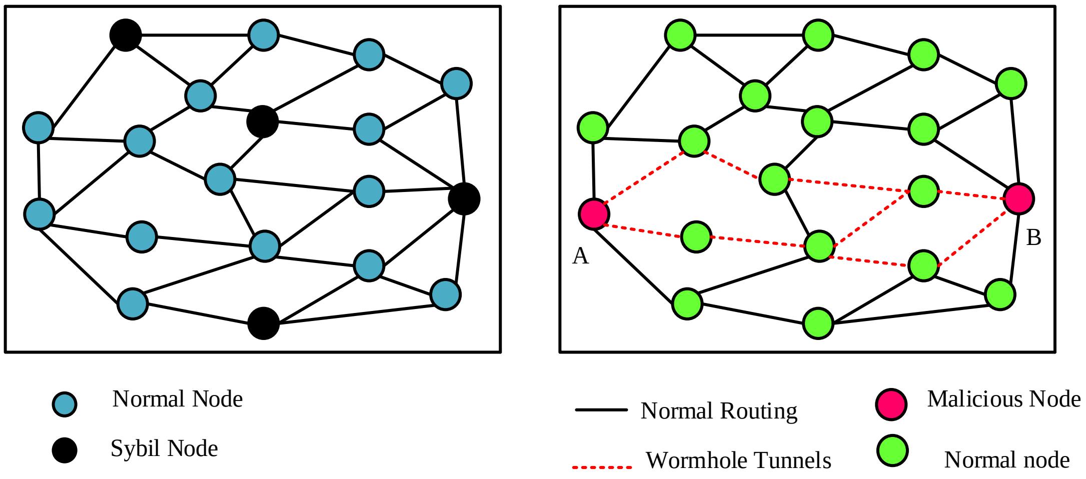


Figure 6.1. Sybil and wormhole routing attacks in WSNs.

6.2 Problem Statement

Wireless sensor networks (WSNs) consist of a large number of autonomous sensor nodes that communicate with each other to monitor and collect data from the surrounding environment.

However, WSNs face several challenges, such as limited energy, computational resources, and vulnerability to security threats and routing attacks. This chapter aims to develop a secure and efficient approach for clustering and aggregation in WSNs using a combination of GA, PSO, fuzzy routing strategy, and hybrid machine learning techniques. The goal is to optimize the network's performance, reduce energy consumption, and enhance security while considering the inherent challenges of WSNs, such as limited resources, dynamic topology, and unreliable communication. The objective of this research is to address the following key challenges:

Design an efficient and secure data aggregation technique within each cluster to reduce the amount of data transmission and minimize energy consumption. The aggregation process should consider data fusion, compression, and eliminating redundant information while maintaining data integrity and accuracy.

Develop an intelligent routing mechanism that utilizes optimization techniques to dynamically adapt the routing paths based on the network conditions, node capabilities, and energy levels. The routing algorithm should consider data transmission's energy efficiency, reliability, and security.

Investigate and employ hybrid machine learning techniques, such as combining supervised and unsupervised learning algorithms, to enhance the performance of clustering, aggregation, and routing processes. The techniques should be tailored to the unique characteristics and requirements of WSNs.

6.3 Network Model

In this model, Wireless sensors are deployed using a hierarchical clustering approach in the targeted environment. The model consists of the sink node, cluster head, relay, intermediate, and ordinary sensor nodes clustered in the network. The cluster head (CH) determines the aggregation of the data flow. There are 1000 sensors deployed in 1000*1000 meter squares with ten clusters in the target field, as shown in Figure 6.2. The sensors are divided into standard and malicious in this traffic network activity. Security techniques are utilized to detect malicious attacks using aggregation and algorithms. The sensor nodes sense environmental conditions, including temperature, pressure, and other changes, and send the data to the cluster head. The cluster head aggregates data and forwards the data to the sink node. It helps route discovery and balances energy consumption. The base station then receives data from the aggregate node. Optimal cluster formation and routing strategy are utilized using intelligent GA-PSO for sensor deployment. This optimized hybrid model helps to minimize cost and energy, optimizing coverage using clustering and routing techniques. The PSO is a technique for finding the optimal solution iteratively. The fitness function is used in PSO with global parameters for the optimal solution. The optimized location of each node is expressed as the average error size of the distance, as shown below in equation (6.1):

$$f_i(x,y)=\left|D_i-\sqrt{(x-x_i)^2+(y-y_i)^2}\right| \tag{6.1}$$

Where (x, y) is of the coordinate of the unknown node, D_i is the distance between any two unknown nodes and (x_i, y_i), denotes the coordinate of the base station location. For finding the optimal position, the fitness function is expressed mathematically by (6.2):

$$\text{fitness}(x,y)=\sum_{i=1}^{n}\left(\frac{f_i(x,y)}{h_i}\right)^2 \tag{6.2}$$

Here, h_i represents the hop size between the base station and the unknown sensor node. The tree-like hierarchical aggregated topology is used in our proposed scheme for WSNs security systems. There are six types of nodes for deploying in the target area: sink nodes, cluster head nodes, aggregate nodes, central monitoring nodes, sensor nodes, and malicious nodes. Every node has its registered unique ID in the sink node. The sensor nodes sense, calculate data and transmit information. The cluster nodes receive data from the sensor nodes and forward it to the base station during aggregation. The sensor nodes as regraded as the leaf nodes used for sensing and forwarding data to the central node. The sensor nodes in the network are required to meet the following assumptions for accurate attack identification. These are:

- Each node must operate as a sensor, base station, or cluster head with specific functions.
- Each node has a link for data traffic control and management capability
- Each node has an index location assignment
- The base station has stationary after deployment for sensor network applications.
- The activation energy is the same for all nodes and is homogenous.
- All nodes must update periodically and measure environmental parameters.
- Each node is adjustable for power transmission by the destination node.
- Sensor nodes distributed in an unattended and harsh environment are not rechargeable.
- The distance between nodes is evaluated based on the distance vector and signal strength.

- The base station receives sensed and aggregated data via the cluster head.

We organize the current secure data aggregation methods by their network architecture. The network model heavily influences sensor nodes' function and the path messages take between them and the base station (BS) [237]. The nodes of some of the preexisting schemes are structured like a tree with its root at BS. The parent nodes in the middle act as aggregators, opening up the option of a multi-hop aggregation path to the BS. The path from sensor to BS is predetermined and exclusive under a tree topology. As a result, the aggregator may become overworked, and its energy depleted over time because dynamic aggregator selection is not fully supported. As a result of the fixed path, information loss is also a possibility. Some techniques employ ring topology or a layered model, in which nodes are grouped into a layer or ring based on their hop count from the BS to reduce the likelihood of data loss.

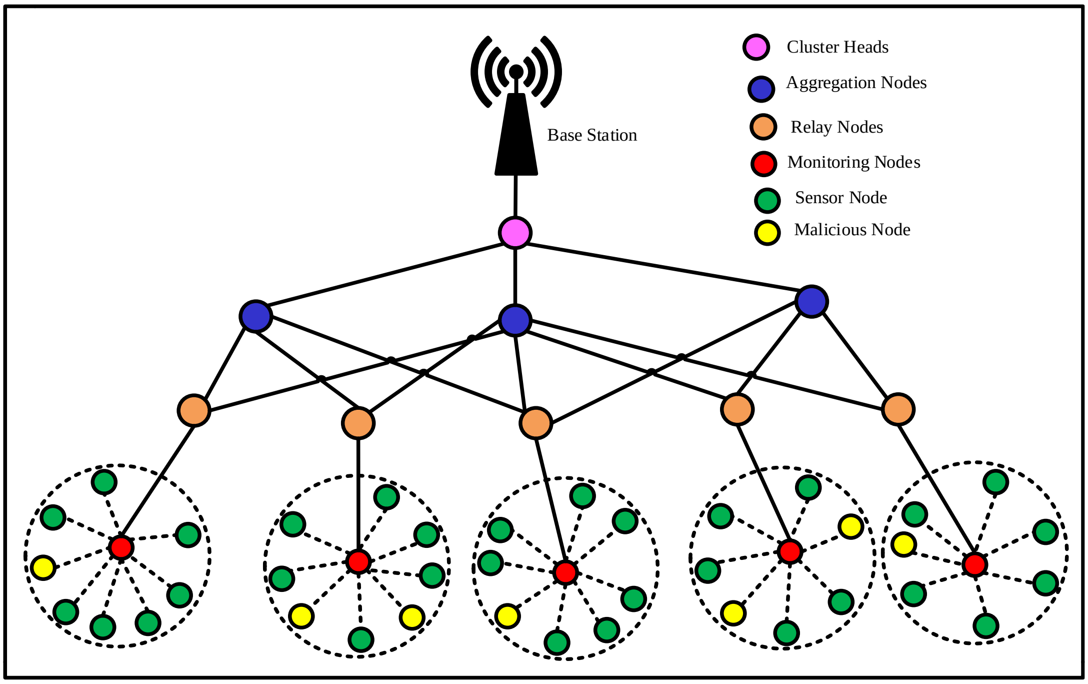


Figure 6.2. Hierarchical aggregated clustered Wireless sensor network model.

The aggregating nodes improve energy depletion, data security, and network lifetime. This model uses the LEACH, AODV, and other security protocols for collaborative security enhancement. These protocols are used for aggregation and topology verification in WSNs clustering and routing. The nodes in the network are assumed to be homogeneous wireless sensor nodes with the same operating power and computational data processing for deployment.

Sensors nodes: Send collected traffic data to a centralized control system.

Monitoring nodes: These manage the data protection and fragmentation in the network.

Relay nodes: Relay nodes regroup and send the fragmented data to the cluster head.

Aggregation nodes: It makes data aggregation and node authentication for sending to the cluster node.

Cluster head: Sends and broadcasts request messages to all cluster nodes for verification.

Base station: Analyzes and checks the received aggregated data for the authentication process.

N sensor nodes are spread out across an X-by-Y-meter monitoring region, with no particular pattern or order to their placement. After being placed in the monitoring region, SNs are supposed to remain in the same locations in the field for the duration of the network's existence. All SNs have GPS systems that allow them to pinpoint their locations in the field. The sink is installed permanently in a location outside of the control room. While SNs are powered by internal batteries, the sink has an unlimited power device to use as much energy as it needs. Each SN is given the same amount of initial battery power (Eo). All of the monitored areas have been chopped up into granules of uniform size. Data redundancy is believed to be produced by all SNs in a granule. Since then, each granule has only had one SN working while the rest have been put to slumber. To create multi-hop connections, two new kinds of nodes-intra-Gateway and inter-Gateway, are added.

At first, the BS executes the clustering and aggregation protocol to use the FCM approach to partition the network into the optimal K number of static clusters. By default, BS chooses CHs for each number of clusters based on fuzzy-based logic rules. Then it uses a hybrid optimization strategy to determine the most efficient route while minimizing energy consumption and cost. The network procedure then repeats itself several times. When necessary, a new set of CHs is elected in a static cluster. Multiple hops and pathways are built from each SN to the sink energy-efficiently utilizing fuzzy rules and sets. Information collected by each SN is sent to CH along the predetermined route to the base station.

Every node gathers the trust information by looking at the behaviour of the near node and the packet forwarded and exchanged. The node also gets information by computing the local trust opinion (LTO) using the packet exchange process [238]. The LTO is computed as in equation (6.3) mathematically:

$$\mathrm{LTO}=\frac{P_{wu}}{P_{wu}+N_{wu}} \tag{6.3}$$

Where P_{wu} and N_{wu}, are the number of positive and negative events for collecting the trust period. The positive events represent successful acknowledgement packets, and the negative events represent unsuccessful packets due to packet drop and packet loss. The trust evaluation parameters for the global trust parameters, including credible reputation, behavioural reputation, subjective reputation, and global reputation, are obtained by the LTO. The Global reputation of node u is computed as shown in (6.4) below.

$$GR_u = \gamma BR_u (1-\gamma) CR_u \qquad (6.4)$$

WhereGR_u, BR_u and CR_u, are the global, behavioural, and credible reputations of the node u. The gamma constant is defined as, $\gamma \epsilon \lfloor 0,1 \rfloor$. The node's reputation is determined by its neighbour nodes' behaviour for identifying the node's status as normal and malicious node primarily with the help of global trust.

6.3.1 Secure Clustering and Data Aggregation

Clustering is a method of organizing a set of sensors that can be used to increase the network's lifespan and reduce power usage [239]. Sensor nodes inside the network are categorized into subsets. Sensor nodes in the cluster collect data, which is then forwarded to a central coordinator. Before sending the information to the hub, the cluster leader aggregates and cleans the data. The clustering approach helps the sensor maintain functionality and accurately evaluate its neighbours. Because of their central role in the monitoring process, cluster heads (CHs) are the most crucial nodes in the cluster. The strategies for selecting a cluster leader are parameterized and can be used for any number of sensor nodes.

- The minimum distance from the base station
- Based on the number of nodes near the cluster head
- The residual energy of the node and the cost of the routing path

This is an illustration of how the distance vector technique is used to calculate the distance vector D between any two sensor nodes as in (6.5):

$$D=\sqrt{\left(x_a - x_b\right)^2 + \left(y_a - y_b\right)^2} \qquad (6.5)$$

The x and y positions of the nodes a and b Any node with a limited distance from the base stations is likely the aggregation node. The cluster's sensor nodes periodically take readings and compile the results for transmission to the cluster hub. The data is compiled and compressed by the CH before being sent to the base station.

Setting threshold settings using the multipath model allows one to assess the energy necessary for network model communication and activation [240]. The amount of energy for k-bits of data transmission over D distance and D_0 threshold distance is given by (6.6):

$$E_{TX} = \begin{cases} k\times E_e + k\times E_f \times D^2, & \text{if } D \le D_0 \\ k\times E_e + k\times E_m \times D^4, & \text{if } D > D_0 \end{cases} \tag{6.6}$$

Where E_{TX} is the transmitted energy, E_f is the reception energy and E_e is the power dissipated in the transmitter or receiver for a single bit of data for transmission. Dissipated energy can be affected by signal propagation, filtering, modulation, and channel coding. The threshold transmission distance d_0 with k length of data, transmission is given by (6.7):

$$D_0 = \sqrt{\frac{E_f}{E_m}} \tag{6.7}$$

For k-bits of message reception, the energy consumed by the receiver node is given by (6.8):

$$E_{RX(k)} = k\times E_e \tag{6.8}$$

The cluster head (CH) is selected among the randomly deployed N number of sensors based on the energy, computational processing, location, and position of the sink node. The set of clusters is defined as Ch ε $\{Ch_{1,}\ Ch_2, \ldots, Ch_j, \ldots, Ch_n\}$. The cluster heads coordinate and monitor the data aggregation and communication inside and outside the group and communicate with the relay nodes, aggregation nodes cluster, monitor nodes, cluster leader head node, and sensor nodes. The Ch has an accurate location and better residual energy to be elected by the base station (BS). The process is formulated by the mathematical optimization problem shown below in equation (6.9):

$$F_{ch} = \alpha\times R_e + (1-\alpha)\times R_1 \tag{6.9}$$

Where F is the fitness function, R_e, is the residual energy of the cluster head transmitted to the relay node and R_l, is the location of the cluster head. The constant α describes the contribution of the energy and location of the cluster head. The clustering protocol has three phases: clustering and aggregation, optimization, data transmission, and evaluation. Optimizing the position and energy improves the network's lifetime. The node near the base station with higher residual power is more likely to be selected as a cluster head [240]. The relay node and cluster head must have more incredible energy and a superior location for energy transmission minimization. The collection of relay nodes is defined so as to minimize the cost between the relay nodes and the cluster head as Rn = $\{R_1, R_2, ..., R_n, ..., R_m\}$. So that the fitness function is by the equation (6.10):

$$F_r = \beta \times R_r + (1-\beta) \times R_1 \qquad (6.10)$$

Where R_r is the residual energy for the relay node and R_l denotes the location of the relay node. A network's clusters, once constructed, don't change at all during normal operation; instead, the topology of the clusters undergoes periodic changes to account for things like growth and shrinkage. The likelihood of each cluster's active SN developing into a CH is estimated with fuzzy sets and a fuzzy rule base. The next CH is chosen among the nodes with the highest potential value. The choice of CH within each cluster is based on two fuzzy-based variables input to the system: the residual energy of the SN and the cost of communication within the cluster. Each input variable is given a normalized value relative to the other variables in its cluster. Normalized and optimal residual energy for i^{th} SN in the cluster can be calculated using the formula (6.11):

$$\text{NorE}(i) = \frac{E(i) - E_{min}}{E_{max} - E_{min}} \qquad (6.11)$$

where E(i) is the energy left over from the i^{th} cluster SN. Minimum and maximum residual energy in this cluster are denoted by E_{min} and E_{max}, respectively.

Secure data aggregation in WSNs ensures data privacy from individual sensor nodes to the central processing units [92]. The integrity and confidentiality of data are two significant concerns in the context of securing data aggregation transmission. Sometimes, the sensor nodes may need to transmit secret information, such as a new set of keys. It is crucial, therefore, in a wireless sensor network to construct a safe communication line. Confidentiality of data means shielding communications from prying eyes. To safeguard sensitive data in a WSN, encryption is employed. The aggregator must decrypt the encrypted data to process it, compromising the data's security. The aggregator may also send incorrect data to the base station by including erroneous information in the aggregate. When information is transferred, it must be verified that it has not been tampered with in any way; this is known as ensuring data integrity.

Data aggregation combines and gathers sensor data using transformation techniques for effective data transmission in the network. It is a systematic approach for data collection from the network's source nodes. Data aggregation extracts useful information and reduces the data using the data aggregation function [93]. Data aggregation is a complementary technique for reducing sensor nodes' network traffic information transmission [241]. The aggregation minimizes the communication costs for information transmission from the sensor node to the sink node by exploiting the raw data correlation. The aggregation technique includes three aspects functional, scheduling, and structural schemes. The aggregating nodes receive from the sensor nodes and combine the data before forwarding secure and aggregated data to the base station. Secure data aggregation enables effective data transmission, power consumption, and improving the quality of services [94]. It requires data integrity and authentication with low cost and high accuracy, provided data privacy is not compromised. Data aggregation and clustering enhance security in WSNs. The aggregated data are prone to compromise attacks, and security issues are significant.

6.3.2 Secure Routing and Communication Strategy

The proposed method offers a novel secure routing algorithm based on a hybrid of the Genetic Algorithm and the particle swarm particle algorithm, which we call hierarchical (GAPSORA). Through a series of calculations, this algorithm determines the most reliable and productive path for data packets to travel from their source to their final destination. This protocol allows efficient routing in WSNs by grouping nodes and establishing a direct path between them. This protocol provides a highly efficient and safe means of routing data. The GAPSORA's advantage is that it reduces node energy usage by employing an intrusion detection mechanism to compute trust levels. In addition, it finds the optimal path with a neural evolutionary algorithm that considers meta-heuristics, trustworthiness ratings, and fuzzy criteria. For WSN to serve its primary functions, it must exhibit data dependability, energy efficiency, and position awareness. The suggested routing approach runs into them because the nodes in this model can cooperate to process the input, and the resulting algorithm provides effective optimal routing. Using an IDS approach to detect malicious attacks and block unauthorized users from participating in the routing and transmission process, the GAPSORA method ensures data packets are sent on time. This algorithm routinely checks the CHs to ensure they reach their target. As a result of this paradigm shift, sensors are equipped with a secure routing system that shields them from security attacks that impede the finding and mapping procedures. The network operates more efficiently since the suggested routing system has a lower packet loss rate.

The proposed protocols' route discovery method is analyzed, and it is dependent on the detection of the attack model and an estimated trust level between neighbouring nodes. By convention, node S is the source, and D is the target in a path calculation. By exchanging RREQ and RREP packets, the optimal path to a destination can be determined. This reduces the time needed to locate a route by finding the optimal and shortest way for routing, showing how long it will take to find a sequence number and all the intermediate nodes that may be used to find a requested message. This data is compiled to develop a secure routing discovery using a fuzzy basis for hybrid routing. There is an initial population of zero in this procedure. Each chromosome has a string of unique integers corresponding to the path's nodes. Each chromosome determines the location of a node along the path. The total number of children in a population is multiplied using a multiplication procedure. It selects only the biological parents from the pool of candidates using the selection operator.

All nodes in a cluster must communicate via CH to function; therefore, if a CH turns malicious nodes for whatever reason, the entire network could be cut off from it. Figure 5 depicts an alternative scenario where CH behaves maliciously in conditions where power consumption is very important to WiMAX networks. Clustering is employed to forestall energy consumption as much as possible. WSN employs an attack detection method called an isolation table (ITID). Some methods use the primary cluster header (PCH) and secondary cluster header (SCH) to spot infiltration (SCH). To divide the SCH's duty cycle, the PCH collects sensing data and an isolation table from the SCH. The SCH keeps an eye out for bad actors inside the MGs and keeps tabs on the PCH. When a cluster's leader is compromised, it has a domino effect on the entire network. The attacker launches an attack on the sensor node to remove CH and alter routing information. When fewer active nodes exist in a WSN, an attacker has an easier time breaking in. As a result, nodes cannot spread the news any faster. If the PCH is under attack, all communications will be at risk.

6.3.2.1 Hybrid GA-PSO Routing Strategy

The genetic algorithm (GA) is the optimization technique inspired by the survival of the fittest stated by the biological Darwinian theory of evolution and reproduction using crossover and mutation operators for gene transfer from parents to offspring [120]. The individuals are selected for mutation and crossover to exchange new traits of mutation and crossover genes. This mechanism enables a genetic algorithm using population as chromosomes represented as binary numbers. The three dimensions of selection, crossover, and mutation are utilized to produce the next generation from the randomly generated population. The objective function evaluates the quality of chromosomes produced in the population applied for optimization using a genetic algorithm. The genetic algorithm optimizes the distance between nodes using the objective function. It also improves the transmission distance and energy consumption by enhancing the network lifetime. The proposed system utilizes clustering and routing using hybrid GA-PSO for the network lifetime, energy consumption, and security performance. The PSO technique optimizes the network coverage, clustering quality, and energy efficiency using optimal clustering and routing. PSO uses a fitness function for evaluating the position of the nodes.

The Genetic algorithm (GA) creates a random P-valued array of candidates for the best answer. In this setting, each possible solution is called a chromosome. An n-dimensional vector with n unique values stands for each chromosome vertices in a network with n edges. In this work, we have chosen the Roulette- Wheel selection approach among the available selection methods in the literature to determine which candidates are qualified and which are not by applying a probability-based equation to the pool of candidates (6.12).

$$P(i)=\frac{F(i)}{\sum_{i=1}^{n}F(i)} \tag{6.12}$$

That is, where F(i) represents the fitness of chromosome I and n represents the population size. Each person is given a value between zero and one on the Roulette-Wheel. A fitness function assigns a value to each chromosome in the population, reflecting its potential for reproductive success. If this value is high, the remedy is effective for that chromosome. Next, the information determines which chromosomes will be included in the solution's offspring. The fitness value is determined by using some fitness function. It is the nature of the problem that has a significant impact on the fitness function.

- Distance, D, to the Sink – depicts the total distance travelled by all sensor nodes (N) to reach the sink node. It is represented by the equation (6.13), indicating how far each sensor ode has travelled.

$$D(N(i),N(\text{Sink}))=D(N(i),CH(i))+\sum_{j=1}^{n}D(CH(j),CH(j+1))+D(CH(n\text{-}1),\text{Sink}) \tag{6.13}$$

Where n is the number of hops from the cluster head CH(i) via CH(j) to the destination node, i.e., the sink node.

- The amount of energy expended while moving all the data to the sink node is denoted by the notation "Transfer Energy" (TE). Here, we use equation (6.14) to determine the energy.

$$\text{TE}(i) = \text{Energy}\left(\textit{node}(\text{i}),\text{CH}(\text{i})\right) + \sum_{\text{j}=1}^{\text{n}} \textit{Energy}\left(\text{CH}(\text{j}),\text{CH}(\text{j+1})\right) + \textit{Energy}\left(\text{CH}(\text{n-1}),\text{Sink}\right) \tag{6.14}$$

The crossover process is the first genetic surgery carried out on the chromosomes of a population. When two chromosomes exchange information, this process is called crossover. Because of this, throughout time, a node at random is selected to act as a crossroads for the various chromosomes. This happens although nodes frequently implicated in both chromosomes are overwhelmingly determined as a potential crossing point. In this manner, the algorithm will discover different paths, some of which may be more productive. Through the crossover process, two chromosomes are joined together by first being selected from the mating pool by applying the selection operator. The genetic material passed down to subsequent generations will be stored on these two chromosomes. A crossover can utilize only a single set of parents during any given process iteration. The chromosomes used in the crossover process have to have at least one node in common with one another, in addition to the transmitting and receiving nodes, for the process to produce effective pathways. When more than one node uses the same connection, the system will randomly select one of them using a certain probability. The node that is determined to be the crossing point is given this designation because it acts as a connection point for two of the edges in the graph.

The chromosomes in the mating pool can then be changed using a mutation operator. There are several potential triggers for a mutation on any given chromosome. It will be disregarded if the outcome of the mutation is the same as any of the chromosomes in the current population. Otherwise, the genetic operator's crossover and mutation have produced a population with a subpar chromosome, which must be replaced with the population's superior chromosome. The procedure described above is iterated indefinitely until the chromosome with the highest fitness value is found to be the most promising strategy for fixing the issue.

The particles, the result of GA's solution iteration, seed the PSO's search space. In order to find the greatest possible answer, each particle remembers its past positions relative to the best particles at the time [242]. The quantified objective function of each particle is recorded. pBest represents the fitness level of the best possible particle at the moment. Consideration of all created populations leads to selecting the optimal value, gBest, from among those values. This work focuses on the optimal path with the lowest possible cost. PSO will attempt to adjust each particle's velocity to move at its pBest. The velocity is based on arbitrary definitions, which produce irrational figures for the direction of travel (pBest). Target value or condition, gBest, and termination value are three global variables that PSO always keeps track of and stores. Each PSO-assessed particle has the following data:

(i) Information that can stand in for the "gBest" global solution.

(ii) The velocity value will represent the required quantity of data modification.

(iii) Top buck.

1. First, we treated all CHs as particles, each with a position and a velocity in two dimensions.
2. The next step is to start generating random solutions. The number of possible outcomes in a random draw is proportional to the population size.
3. At this point, we'll use the fitness function, minimal path distance, to estimate the fitness value.
4. New particles are created by sampling the initial distribution of possible outcomes. The creation of a new particle involves the breakdown of an existing particle into its constituent parts.
5. To assess newp's fitness, we use the path's distance as a proxy.
6. The fitness values of the old and new particles are compared, and the one with the highest value is taken forward to the next cycle.
7. At the end of each cycle, the best solution is chosen as the pBest. This iteration chooses the particle with the highest fitness value as the pBest solution.
8. The gBest solution is chosen from the set of pBest solutions found throughout all particle iterations, where is the maximum value among all solutions. At last, the approach for gathering data from different clusters that yields the gBest solution is chosen.

6.3.2.2 Hybrid ACO-PSO Routing Strategy

The proposed system utilizes a hybrid ACO-PSO to improve the network's performance and lifetime. Ant colony optimization is a Meta-heuristic technique for inspiring ant colony behaviour. The ACO technique finds a routing path from the sensor node to the sink with alpha and beta-generated values. The generated results are given to PSO based on refinement to find the best and shortest routing path. The ants follow the shortest path to detect the system phenomenon.

Similarly, the nodes in the proposed system find the optimal route for sending data to the cluster head. Particle swarm optimization is a strategy for the ideal rearrangement of particles by changing the direction using stochastic and deterministic segments. Every particle is affected by its position, and its vector and speed characterize it.

The behaviour of ants inspires ACO to find the shortest path between their nest and food sources. In the context of the proposed system, ACO is used to find a routing path from the sensor node to the sink or cluster head. The algorithm considers the pheromone trails (alpha) and heuristic information (beta) to guide the ants in their path selection. The generated routing paths from ACO are then passed to PSO for further refinement.

PSO is a technique that mimics the social behaviour of a swarm of particles, such as birds or fish, in searching for the optimal solution. In the proposed system, PSO is employed to find the best and shortest routing path based on the results generated by ACO. Each particle in PSO represents a potential solution, and its position and velocity vectors define its movement in the search space. The particles adjust their positions based on their own best-known solution and the global best-known solution within the swarm.

The proposed system aims to achieve an optimal routing path for data transmission in the network by combining ACO and PSO. The ACO algorithm initially guides the ants to explore the search space and find potential paths, while PSO further refines these paths to identify the best solution. This hybrid approach allows the nodes in the network to find the optimal routes for sending data to the sink or cluster head, thereby improving the network's performance and extending its lifetime.

6.3.2.3 Communication Strategy

The BS starts by locating the CH nodes in each cluster. Until the CH's remaining energy drops below the cluster's threshold energy, it will continue to serve as CH in subsequent rounds [243]. This method conserves resources by eliminating the constant need to form CHs. The cluster's threshold energy is continuously evolving in preparation for a subsequent CH election procedure. When the current cluster head's residual energy falls below the cluster threshold, a new CH election process is triggered, and the cluster enters an active-sleep state. Each cluster's active SNs survey the other SNs dormant in their granule and switch on the one with the largest residual energy. The current CH then uses the suggested hybrid fuzzy inference mechanism to select a new CH from the cluster's newly-active nodes. It sent out a signal to all of the operational CMs in the cluster.

Cluster heads (CHs) are responsible for collecting data from their cluster members (CMs), aggregating it, and then sending it to the base station (BS) in a single hop or several hops. On the other hand, CMs only need to make one hop to transmit their sensed data to the CH. As a direct consequence, CH nodes are responsible for more work than CM nodes. Within the same cluster is a specialized type of node called an intra-cluster gateway (GW) node. This type of node is responsible for most of the work the GW node would normally perform. If given the opportunity, CH will select the path of BS that will provide them with the greatest amount of untapped potential energy. The CH of the cluster distributes a TDMA schedule to every node in the cluster. When the given time has elapsed, the CM will transmit the data it sensed to CH. The next step is for CH to gather all of the information and then use the CDMA protocol to transmit it to the GW node that has been identified. If a GW node is selected that is located inside the designated virtual region, it is possible for there to be direct communication between the GW and the BS. If this is not the case, messages must be passed between clusters. When the BS is located outside the monitoring area, the GW nodes spend disproportionate energy on communications due to the long distance between themselves and the BS. This study proposes a method of avoiding this problem by using fuzzy rules and fuzzy sets in multiple communication paths. This proposed scheme also utilized a fuzzy inference system for determining optimal multiple-hop routes. A GW node within the cluster uses a fuzzy inference approach to locate a sensor node in a cluster that is not currently connected to its own. An inter-cluster GW node is the one that has been chosen.

6.4 Attack Models

Network traffic analysis eavesdrops on the data transmission in WSNs. The attacks are divided into layers in WSNs based on the attack behaviour. The traffic analysis of the data can be converted into a suitable format for evaluating the attack detection performance. The malicious attacks in WSNs compromise the nodes by creating fake identities, such as Sybil attacks, and creating tunnel-like wormhole attacks, as shown in Figure 6.3. Sybil attacks duplicate multiple malicious nodes with fake identities affecting the network's data aggregation. This makes the busy the aggregator nodes and base station. In contrast, the sinkhole attack exploits the network resources by advertising itself to the neighbour node as a normal node. The sinkhole attack attracts the healthy node and drops the packets that affect the network layer's data aggregation.

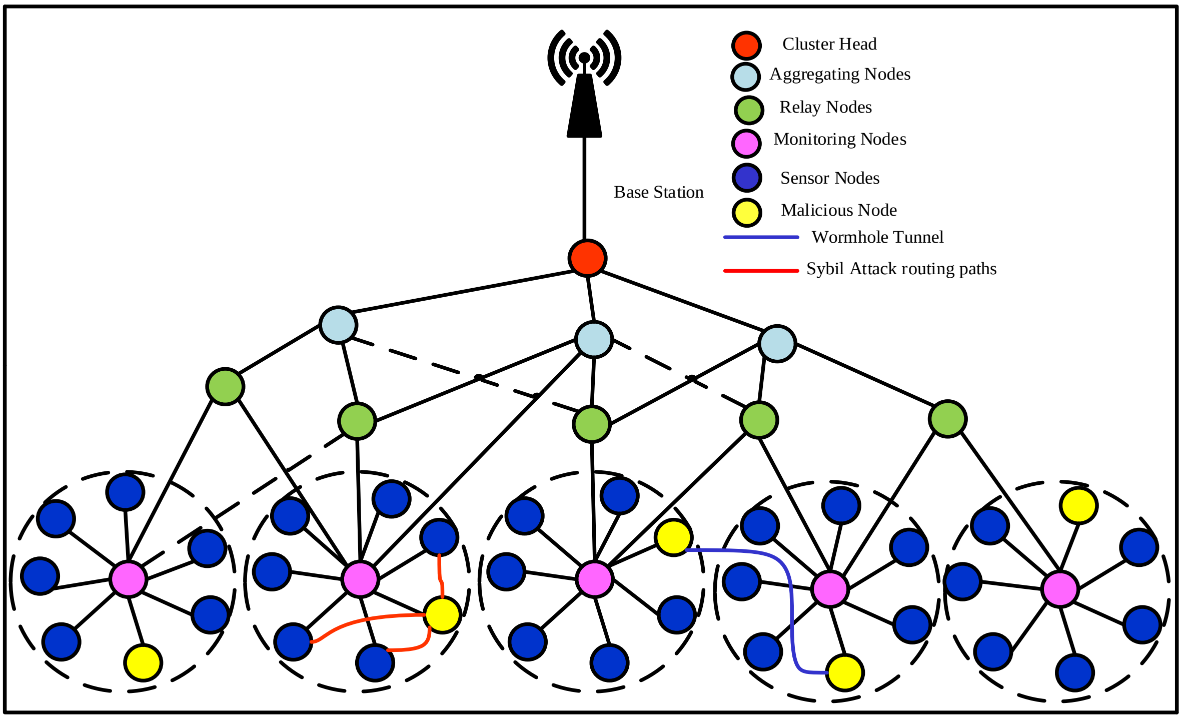


Figure 6.3. Proposed wireless sensor network attack model with Sybil and wormhole attacks.

Some methods are employed to defend the sensor node from assaults, such as selective forwarding, sinkhole attacks, spoofing, black holes, and Hello flood attacks [124]. Sensor networks encrypt their internal data via the advanced encryption standard (AES), the RC5 method, and the Skipjack approach, which relies on previously dispersed keys to prevent attacks like eavesdropping and traffic analysis. Faulty nodes can be isolated using a linear autoregressive predictor, and their damaging behaviour can be remedied if possible. Rogue nodes can also be identified by localization anomaly detection. There is also the possibility of using the strength of the signal to identify a rogue node. The primary purpose of attacks on time synchronization protocols is to trick certain nodes into thinking that their neighbours' clocks are set to a different time than they are. It is important to note that de-synchronization attacks are also transport layer attacks. A cluster is a collection of nodes created into a single entity through clustering. The primary goal of forming clusters is to lessen the frequency with which members are transferred to new groups. The CHs use simple principles to detect numerous types of route attacks. Since the CHs are chosen randomly, a hostile attacker may get elected as a CH and harm the network.

6.5 Proposed Framework

The proposed system includes intelligent sensor deployment, clustering and aggregation, data processing, training and testing, and attack detection and classification. A group of nodes that work together to detect and eliminate spoofed data before it reaches the cluster master and is sent on to the network's foundation. This improves the ability to identify and isolate rogue nodes by strengthening their authentication [94]. Secure data aggregation in hierarchical and clustering WSNs is seen in Figure 6.4. The cluster's leader comprises data such as node identifiers and messages. The cluster's monitoring and relay nodes will choose the best path for sending data to the cluster's primary node. Nodes can only enter or leave the cluster at the discretion of the monitoring node. The information is checked at the aggregation node before being sent to the sink node, which identifies and categorizes attacks. Safe data aggregation and clustering enhance networks' reliability, longevity, and computational power available to users [191].

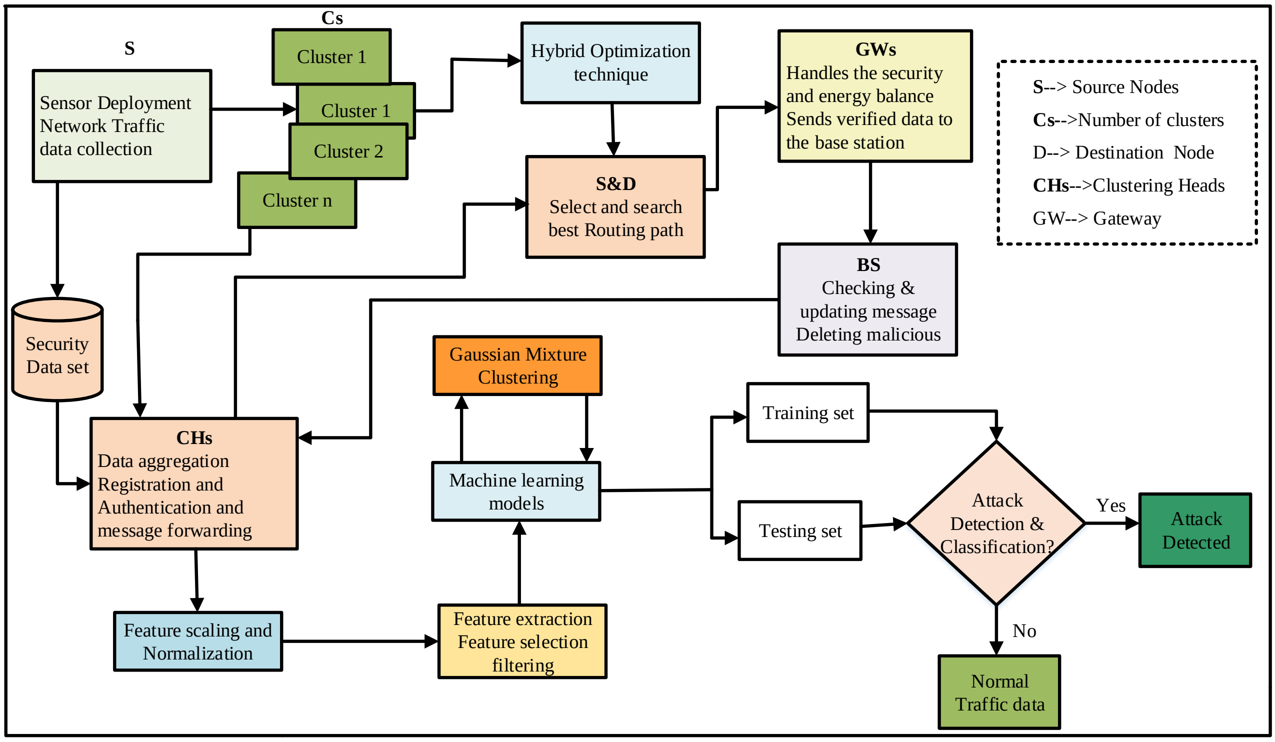

S
Sensor Deployment Network Traffic data collection
Cs
Cluster 1
Cluster 1
Cluster 2
Cluster n
Hybrid Optimization technique
S&D
Select and search best Routing path
GWs
Handles the security and energy balance Sends verified data to the base station
S--> Source Nodes
Cs-->Number of clusters
D--> Destination Node
CHs-->Clustering Heads
GW--> Gateway
BS
Checking & updating message Deleting malicious
Security Data set
CHs
Data aggregation Registration and Authentication and message forwarding
Gaussian Mixture Clustering
Machine learning models
Training set
Testing set
Attack Detection & Classification?
Yes
Attack Detected
No
Normal Traffic data
Feature scaling and Normalization
Feature extraction Feature selection filtering

Figure 6.4. Proposed secure clustering and aggregation in WSNs based on machine learning.

The proposed work is a hybrid model with various features for an effective attacker detection process. The structure of the proposed system is shown in Figure 6.4. The simulation units materialize the system's determination and dynamic design in a wireless sensor network to deploy the real-time system. The proposed hybrid security system has different phases for different attack classes of the malicious activity of the network.

The proposed scheme has several units for detecting and classifying attacks using hybrid techniques. The main phases of the proposed system are intelligent deployment unit, database and processing, training and testing phases, and attack detection and classification.

6.5.1 Intelligent Sensor Deployment Routing

Figure 6.5 depicts one of the phases of the suggested methodology: setting up the network, collecting and routing the data, preprocessing the data, feature extraction, and finally, intrusion detection [46]. Data preprocessing was the first analysis after the dataset was acquired and loaded [139]. Preprocessing data is crucial since it helps eliminate anomalies and duplicate information. The sensors are deployed using the target field's intelligent hybrid routing technique. Sensor nodes detect and collect information for sending to the cluster head for aggregation. The cluster head (CH) aggregates and forwards the data to the base station. The CH also manages resource utilization by reducing the data for transmission to the sink node. The wireless sensor nodes are randomly distributed for sensing and collecting data.

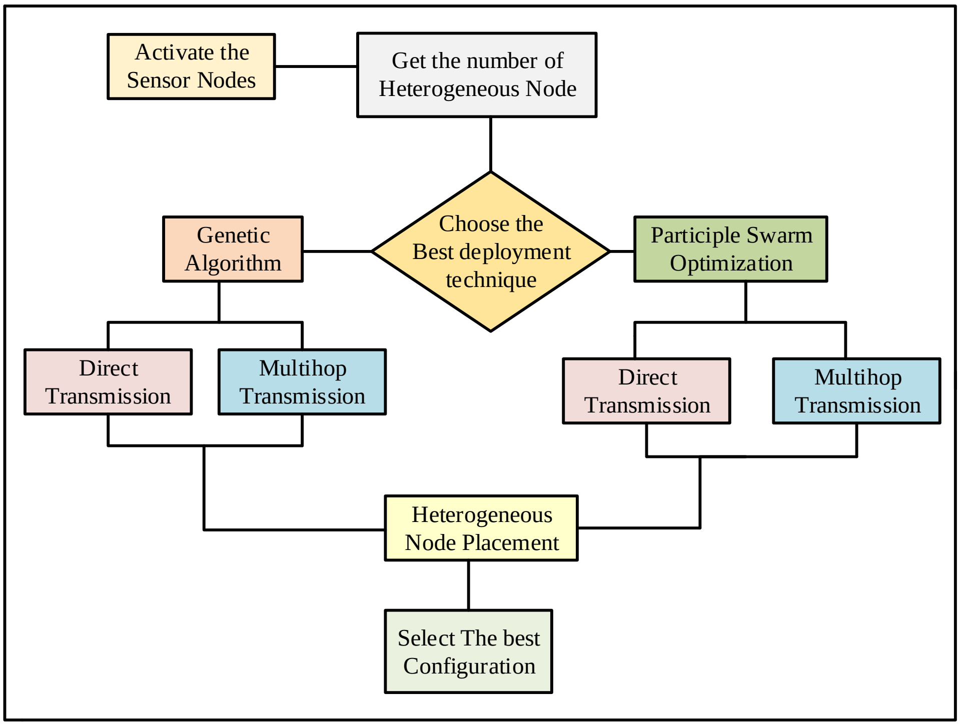

Activate the Sensor Nodes
Get the number of Heterogeneous Node
Genetic Algorithm
Choose the Best deployment technique
Participle Swarm Optimization
Direct Transmission
Multihop Transmission
Direct Transmission
Multihop Transmission
Heterogeneous Node Placement
Select The best Configuration

Figure 6.5. Hybrid GA-PSO in WSN routing for sensor deployment with direct and multihop routings.

The nodes in the network model are categorized into sensors, cluster heads, and sink nodes. The deployment unit in the proposed system is intelligent, using genetic and particle swarm optimization techniques. Direct and multi-hop routing techniques are applied for effective wireless sensor network planning and design. The planning and deployment technique is based on the hybrid GA-PSO technique using the direct and multi-hop transmission schemes, as shown in Figure 6.5. The wireless sensor nodes are assumed to be heterogeneous with different data and computational processing.

6.5.2 Optimization Techniques

Optimization addresses the nondeterministic polynomial complex problem using an objective function that minimizes or maximizes the problem [120]. Optimization techniques are metaheuristic techniques that use high-level procedures. The objective function in the optimization technique can be a single or multi-objective function in which all the points converge to a single optimal solution and point. Particles can connect to multiple optimal solutions in multi-objective functions. Optimization techniques are utilized to create well-functionally network design and planning to achieve security and network lifetime. The proposed system uses hybrid routing to enhance WSNs' security performance and optimal energy consumption for achieving the desired network lifetime.

Optimization techniques are crucial in enhancing the performance, efficiency, and security of wireless sensor networks (WSNs) integrated into the Internet of Things (IoT). These techniques aim to optimize various aspects of the network, such as resource allocation, energy consumption, routing decisions, and data aggregation. Here are some commonly employed optimization techniques in the context of WSNs:

Energy Optimization: Energy-Efficient Routing: Algorithms are designed to find energy-efficient paths for data transmission, considering factors such as node energy levels, transmission distances, and channel conditions. Nodes are put into sleep mode when not actively participating in the network, reducing energy consumption. Sensor node power usage is optimized by dynamically adjusting the operational modes of components like sensors, radios, and processors based on workload or environmental conditions.

Resource Allocation: Techniques are employed to optimize the allocation and utilization of available spectrum resources, mitigating interference and maximizing network capacity. Algorithms are also used to allocate bandwidth efficiently among sensor nodes based on their communication requirements and priorities. Time-division multiple access (TDMA) or other scheduling algorithms allocate time slots effectively for sensor nodes to transmit their data.

Quality of Service (QoS) Optimization: Routing protocols are designed to consider QoS metrics such as delay, reliability, and bandwidth requirements when determining the optimal paths for data transmission. Prioritization mechanisms differentiate traffic based on their importance or latency requirements, ensuring that critical data receives preferential treatment.

Security Optimization: Efficient key management schemes are employed to optimize the distribution, refreshment, and revocation of cryptographic keys, ensuring secure communication among sensor nodes. Optimization techniques are used to efficiently identify and respond to intrusion attempts, minimizing false positives and negatives in intrusion detection systems. Techniques are employed to optimize data aggregation while maintaining data integrity and privacy, reducing the amount of transmitted data and conserving energy.

Localization Optimization: Techniques are utilized to optimize the accuracy of node localization, minimizing errors and improving the overall positioning performance in the network. Algorithms are designed to optimize the localization process regarding time, energy consumption, and resource utilization.

Traffic Optimization: Compression algorithms are employed to reduce the size of data packets, minimizing bandwidth utilization and energy consumption during data transmission. Aggregation techniques are optimized to reduce redundancy and eliminate unnecessary transmissions, saving network resources and improving efficiency.

6.5.3 Benchmark Datasets

The actual traffic scenario is reflected by utilizing a cyber-security dataset containing regular network DoS attacks to represent real-time network traffic [51]. The dataset consists of the behaviour of the users using the protocols to simulate an accurate real network environment. After data is acquired, the dataset is stored in the cyber security database for the next section. The dataset is stored in different formats, containing much information. The primary task of machine learning approaches is performing classification tasks based divide and conquer strategy with discrete units.

6.5.3.1 NSL-KDD Dataset

The NSL-KDD Dataset is used as a benchmark for evaluating the performance of the proposed system containing sub-datasets, as shown in Table 3 and 43 features per record [144]. Two datasets, NSLKDD and UNSW-NB15, are used for experimental analysis. New and upgraded from the KDD 99 cup dataset is the NSL KDD dataset [46]. The NSL KDD dataset includes symbolic, numeric, and Boolean features of varied resolutions and ranges. All observable characteristics were utilized in this analysis [220]. The training and testing sets in the current NSL-KDD dataset have 148517 connection records and the same number of attributes. Using a cross-validation method, we split the data into two sets: 80% for training and 20% for testing. The two features are labels for normal and abnormal traffic networks, and the 41 features are the traffic to the input. The NSL-KDD Dataset consists of four attacks: Dos, R2L, U2R, and Probe. The dataset is partitioned into four samples with respective numerical distributions, as shown in Table 3. The dataset is divided into 80% of training and 20% of testing, as shown in Table 6. 1.

Table 6. 1. Frequency distribution of attacks in the NSL-KDD Dataset benchmark dataset.

Dataset	Number of traffic records in the network					
	Total	Normal	DoS	Probe	U2R	R2L
KDDTrain+20%	25192	13449	9234	2289	11	209
KDDTrain+	125973	67343	45927	11656	52	995
KDDTest+	22544	9711	7458	2421	200	2654

We evaluate the proposed system's ability to recognize and categorize attacks using machine learning models on the NSL-KDD, CICIDS2017, and UNSW-NB15 benchmark datasets. This study presents a dataset for intrusion detection and network attack scenarios, which was gathered and run using tools. Common and typical types of attacks based on real-world facts, such as network traffic monitoring, are included in the dataset. The dataset is highly diverse, including 80 feature sets derived from CICFlowMeter assaults of varying types.

The NSL KDD dataset includes symbolic, numeric, and Boolean features of varied resolutions and ranges. All observable characteristics were utilized in this analysis [220]. The training and testing sets in the current NSL-KDD dataset have 148517 connection records and the same number of attributes. Using a cross-validation method, we split the data into two equal sets: 80% for training and 20% for testing. There are a total of 42 features in this data set. The last of these features (42) is a class feature. In this work, the feature engineering strategy is used for feature extraction. Nineteen features from the NSL-KDD dataset, with eighteen features extracted from the UNSW-NB15 dataset, are chosen using the suggested method.

The proposed study uses the CSV-formatted NSL-KDD intrusion dataset for testing and evaluation [135]. The NSL-KDD Dataset is utilized to evaluate the performance of the proposed system. It has multi-class data having 41 features about the characteristics of a particular type of attack on the network. This dataset consists of 34 numerical features and seven character features, and it covers "4" primary classes as indicated below: DoS, Probe, R2L, and U2R.

- With a probe, an attacker tries to learn more about the victim's network by scanning the latter's servers and other devices.
- Attacks like Denial of Service (DoS) render networks or computers unreachable to their intended users.
- With a User to Root (U2R) assault, an adversary attempts to convert a regular user into a system administrator.
- An R2L attack is one in which the attacker controls the victim's system throughout the whole network without the victim's knowledge.

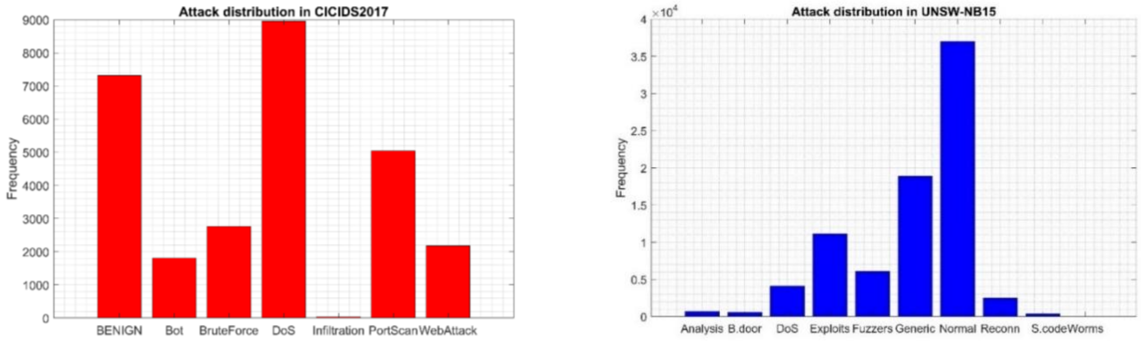
The dataset has 41 features, including the protocol, duration, service, source bytes, flag, destination bytes, etc., with 80% considered for training and 20% for testing. The dataset has 38 numeric and three categorical features. The statistical distribution of the dataset is shown in Table 3. 9.

Table 3. 11. Frequency distribution attacks in the dataset.

Class of attacks	DoS	Normal	Probes	R2L	U2R	Total
Frequency distribution	45927	67343	11656	995	52	125973
Percent (%)	36.5	53.5	9.3	0.8	0	100

6.5.3.2 UNSW-NB15 Dataset

The UNSW-NB15 is a state-of-the-art intrusion detection benchmark dataset used in numerous recent studies and earlier efforts. The Australia Center for Cyber Security's cyber range laboratory IXIA PerfectStorm was used to produce the raw data traffic packets (ACCS) [151], combining "regular" network packets with "abnormal" ones. The software continuously updates the vulnerabilities and exposures of collected packets in the context of information security, simulating nine different types of assaults. With three types of category characteristics and 39 numerical features, the dataset has a total of 42 properties. The dataset contains ten different types of attacks, and their relative prevalence is shown in the accompanying diagram as in Figure 6. 6(b).



(a) Distribution of assaults based on their frequency in CICIDS 2017.

(b) Distribution of attacks according to the frequency in UNSW NB15.

Figure 6. 6: Frequency distributions of attacks were plotted using the CICIDS2017 and UNSW NB15 benchmark datasets.

The dataset is split into training and testing sets to train machine learning models to make accurate predictions in classification and regression.

6.5.3.3 CIC-IDSS2017 Dataset

The CIC-IDSS2017 dataset is used as a benchmark for evaluating the performance system for detecting flooding attacks in WSNs. The dataset is available online at the Canadian Institute for cyber security research LAB. The dataset includes seven classes of attacks, as in Figure 6. 6 (a), according to the attack sample frequency distribution. The dataset contains benign and other attacks that include network traffic data [132]. It was generated for realistic background traffic for building the dataset. The dataset was created using 25 users with different protocols. The dataset contains 485881 Instances and 31 attributes with 80% of training and 20% of testing. Five denial of service (DoS) attack classes include Slowhttptest, slowloris, Hulk, Heartbleed and GoldenEye [133]. The structure of the dataset is shown below in Table 3. 10.

Table 3. 12. Structure of the dataset with classes of attacks.

Attacks	Normal	slowloris	Slowhttptest	Hulk	GoldenEye	Heartbleed
Training	252382.6	1504.8	1276.8	127302.4	6307.2	8
Testing	63076.4	376.2	319.2	31825.6	1576.8	2
Total	315382	1881	1596	159128	7884	10

Processing is utilized to rearrange the data via normalization, missing value imputation, and aggregation for both the training and testing phases. The gaps in data are filled in using the current values' averages [134]. Data is converted from their original format into binary digits using the minimum and maximum values.

6.5.4 Dataset Pre-Processing

The data processing begins by cleaning the original dataset and formatting it suitable to the model input. The original dataset is in pcap format and CSV files detailing the features of the traffic data using Wireshark. The dataset is transformed into model input format as shown in Figure 6. 7. The feature that contains the attack labels is converted into 0 and 1 for normal and abnormal attack categories.

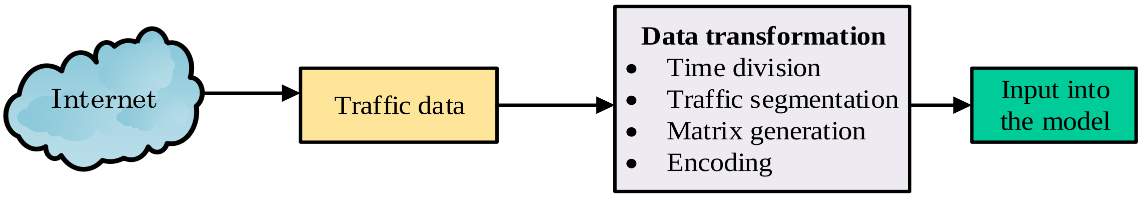


Figure 6. 7. Data processing block diagram for training and testing

The time division intercepts the pcap file period from the original pcap file about the attack time and type. The pcap files are further segmented into sessions according to the IP of the attack host and victim host. The pcap files are also encoded into a fixed-length matrix for the input model. The data processing unit contains the activity including:

- Feature selection and extraction using the decision tree repressor Scikit-learn library for statistical analysis as in Figure 6. 7.
- Remove redundant features and datagrams from the dataset.
- Feature Transformation using principal component analysis for optimal dimensions and variance.
- Clustering and maximizing essential features using the Gaussian mixture model.
- Checking the dataset labels is also part of the process.

The principal component analysis (PCA) is used to reduce the dimensionality of the dataset by compressing and filtering noisy data at all the node levels, as shown in Figure 6.8. It also reduces the cluster head communication overhead and buffer overflow. It applies to fault detection and localization, data aggregation, and tracking in WSNs. The K-means clustering is also applied for dividing the Dataset into K clusters. It is also useful for selecting the best cluster heads in routing in WSNs.

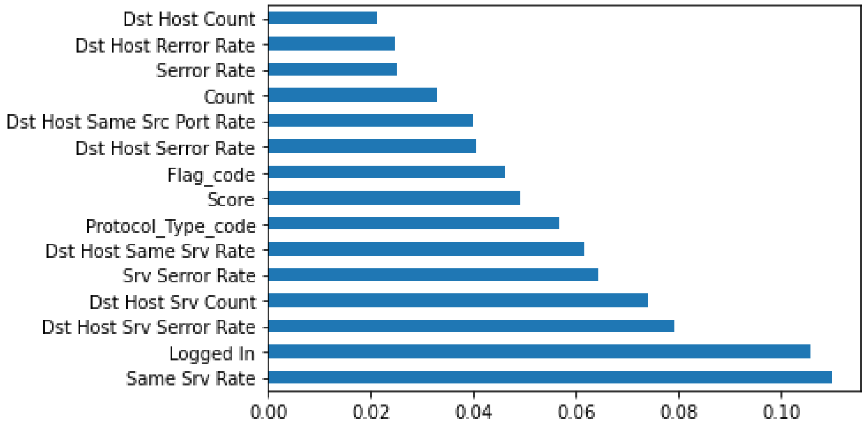


Figure 6.8. Selected input features before modification using clustering.

Two datasets, NSLKDD and UNSW-NB15, are used for experimental analysis. New and upgraded from the KDD 99 cup dataset is the NSL KDD dataset [46]. The NSL KDD dataset includes symbolic, numeric, and Boolean features of varied resolutions and ranges. All observable characteristics were utilized in this analysis [220]. Using a cross-validation method, we split the data into two sets: 80% for training and 20% for testing. There are a total of 42 features in this data set. The last of these features (42) is a class feature. As a bonus, the UNSW-NB15 dataset was developed using the IXIA Perfect Storm tool in the Cyber Range Lab at the Australian Centre for Cyber Security (ACCS). This data set is a mixture of everyday actions and malicious ones. Among its 49 characteristics is a label indicating its classification. In this work, the feature engineering strategy is used for feature extraction. Nineteen features from the NSL-KDD dataset, with eighteen features extracted from the UNSW-NB15 dataset, are chosen using the suggested method. For this data, we preprocessed it using Normalization and the Min-max methodology. The resulting minimum-maximum value is then used in a PCA method to pick features. Using PCA, we narrowed down the 49 attributes in the dataset to the most salient components.

6.5.5 Machine Learning Techniques

Random forest, REPTree, Naive Bayes, J48 Tree, minimal sequential optimization, and bagging are then used to train on the smaller dataset. Data is collected and routed through wireless sensor networks in the first stage. In the pre-processing phase, the datasets are passed on to have any duplicate or erroneous data removed. The data are then sent to the feature selection phase after preprocessing. In the feature selection process, we pick out the most relevant aspects of the dataset. The fast correlation filter is a linear correlation coefficient used to improve the quick correlation and feature selection algorithm, which is then applied to the feature selection process. Hybrid machine learning models are fed the selected features for training and testing to aid in detecting and categorizing intrusions.

6.5.5.1 Naive Bayes

Naive Bayes is a basic technique for classifying data based on probability theories to identify which classes should be included. Predictions can be made after just one scan, and the process is straightforward. The technique is predicated on a streamlined version of the Bayesian theorem.

Conditional probability theory is used to predict to which class a dataset sample will belong. Samples from the test dataset have their classes determined based on the system's prior training on the training dataset. Despite its seeming lack of complexity, the Naive Bayes algorithm is highly effective. The mathematical formulation for the probabilities involved in Bayes' theorem is shown in (6.15):

$$P(c|x_1,x_2,....,x_n)=\frac{P(x_1,x_2,....,x_n|c)P(c)}{P(x_1,x_2,....,x_n)} \tag{6.15}$$

Where, P(x) is the probability of event x, c is the desired outcome, and x is the entire dataset's properties.

Based on Naive Bayes, an optimal cluster head selection method is utilized for safe and low-power routing in WSNs. An ideal collection of CHs will always maximize the network's lifetime while minimizing the energy drain on individual sensor nodes. Naive Bayes ensures continued network flexibility in the face of dynamically added or modified features. This study describes a new adaptive integrated routing architecture for data collecting using a Bayesian approach.

6.5.5.2 Logistic Regression

Logistic regression (LR) is a statistical model for a predefined list of data categories by estimating the probability of the class of attacks using the logistic function [187]. Logistic regression I designed for solving classification problems using the logistic regression function. It has two statistical models for logistic classification: linear regression and logistic classification. The first uses a fitting curve to predict continuous values for training the data, and the second one predicts discrete values for the classification of the training samples. Logistic classification is defined mathematically as in equation (6.16):

$$h_\theta(x)=\theta^T x \tag{6.16}$$

x is the input vector containing input features, and θ is the optimization model parameters utilized for training samples. The decision boundary for linear and nonlinear application of the logistic classification is defined by equation (6.17):

$$f(z)=\frac{1}{1+e^{-z}} \tag{6.17}$$

Where z is defined as:

$$z=\theta^T x \tag{6.18}$$

The sigmoid function measures the certainty level of the new observation in data samples for classification. It is continuous and monotonically increasing, probably represented by f(x). The logistic regression estimates the discrete values in 0 or 1 using independent values by fitting the threshold value. Logistic regression has critical features, including low computational burden, well-calibrated prediction probabilities, and correlated features. It is also helpful for predicting and forecasting the relationship between two variables. It is applied for localization and data aggregation in WSNs.

6.5.5.3 Random Forest

When several decision trees are trained using many different data sets, the resulting algorithm is called a random forest algorithm (RF). Breiman created this multi-technique classifier back in 2001 as an algorithm. Sub-training clusters are generated in the random forest algorithm. When forming a training cluster, preloading is used. To grow the trees, we employ a mechanism in which the attributes are randomly picked. The algorithm works by picking a random value from each node and utilizing that as the basis for a branch. Randomly selected factors produce the derived trees. The collected datasets are utilized as input into the Classification and Regression Trees (CART) algorithm for tree building. Each created tree is used to label the training sample and the classes assigned to which the sample is then compiled. To be processed instances are often included in the most common classification to which they belong. The RF method does not include pruning, while the CART algorithm does. An important reason why the RF algorithm outperforms the other decision tree approaches is that it doesn't rely on pruning. The RF algorithm is quick, flexible, and more effective than alternative decision tree approaches despite its usage of numerous tree topologies. The CART algorithm uses the GINI index value to decide which branch to create from each node. Tree development parameters include the number of trees and the number of variables per node. The RF algorithm's basic operation is depicted in Figure 6. 9 for attack detection and classification using training and testing the benchmark datasets.

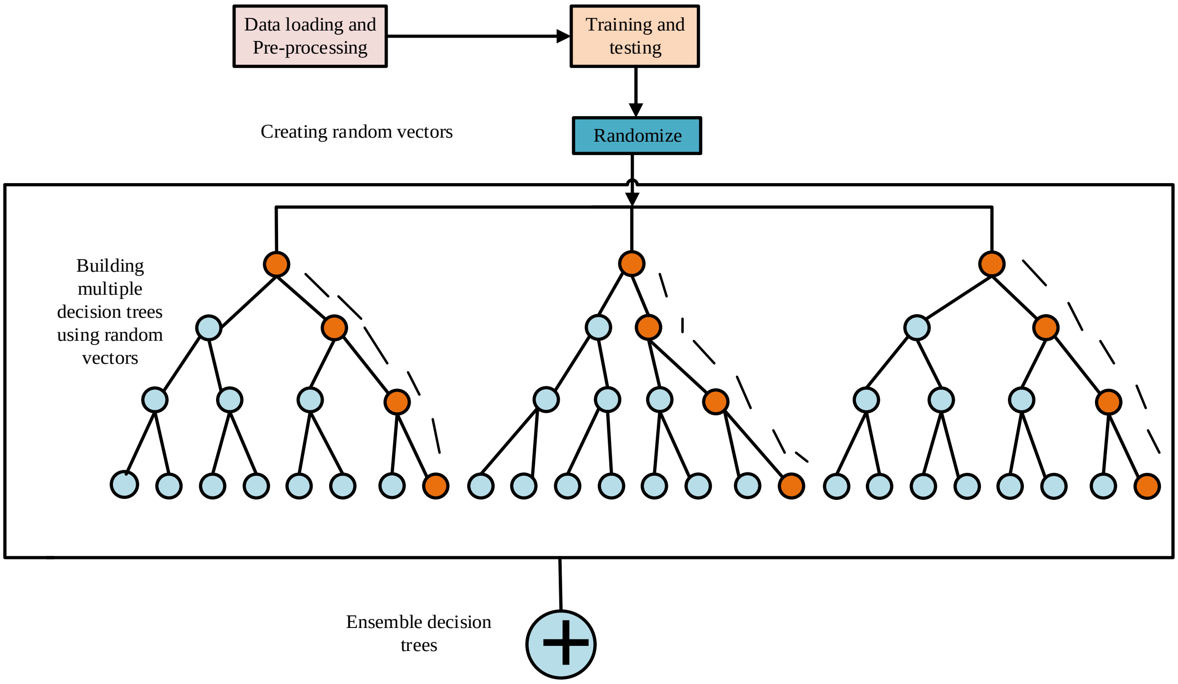

Data loading and Pre-processing
Training and testing
Creating random vectors
Randomize
Building multiple decision trees using random vectors
Ensemble decision trees

Figure 6. 9: Block diagram of random forest operation for training and testing benchmark dataset [38].

Random forest (RF) is a machine learning classifier consisting of a group of trees for building a classification predicting model [119]. The RF technique is effective for two steps, creating a random forest classifier and predicting results [142]. RF technique is effective for a large and heterogeneous dataset to accurately predict missing values. Random forest randomly selects training samples and isolates variables for each tree node to produce more decision trees. Random forest is an appropriate classifier for hyperspectral data for solving coverage and medium access control (MAC) protocols in WSNs. It is robustly applied for highly correlated data and high-dimensional data. When dealing with high-dimensional data, Random Forest handles it by evaluating the significance of features to solve overfitting and stability issues by lowering variance [136]. Because of this, the random forest method is useful for identifying and categorizing malicious assaults in wireless sensor networks utilizing a diverse benchmark dataset. Also, the random forest can deal with missing data and is resistant to anomalies without too much impact on the rest of the data. Due to the extensive tree-building involved, the plan also necessitates greater computing power and additional resources. To forecast the detection accuracy of the model, Random Forest builds a forest out of many decision trees [143]. It can recognize and classify threats using a common attack dataset.

6.5.5.4 REPTree

REPTree is a machine learning classifier for reducing error-pruning trees for building decision-making models. It is a decision tree built with information variance obtained and uses a reduced error pruning technique for error minimization [244]. The numeric values of the attributes are conquered into subsequent pieces of samples. REPTree is a machine learning model for decision and prediction-making processing. The REPTree uses regression logic and creates multiple iterations for measuring mean square error for building predictions [245]. Constructing regression trees uses information gain and is the quickest decision tree machine learning model.

The algorithm starts with a full decision tree that classifies the training data perfectly. It then evaluates the quality of each node in the tree using a splitting criterion, typically based on information gain or Gini index. If splitting a node leads to a decrease in classification accuracy, the node is considered a leaf node, and the decision tree is pruned. This process is repeated until a stopping condition is met, such as a minimum number of instances per leaf or a maximum tree depth. The first step is to divide the training data into subsets based on a characteristic chosen at the root node [246]. In doing so, we make a partition for each conceivable value of the feature. The characteristic splitting and the root node's range increase linearly with increasing information gain. By dividing the training dataset (Y) into smaller subsets (Yi), we can calculate the information gain (IG) as in equation (6.19):

$$IG = -\sum_{i=0}^{N} \frac{|Yi|}{Y} E(Yi) \tag{6.19}$$

The operator |·| is the scope of the set, and E(Yi) is the entropy statistic of the set Yi, defined as (6.20):

$$E(Yi) = -\sum_{j=1}^{N} Pj \log_2 \left(Pj\right) \tag{6.20}$$

N is the total number of sleep stages, and pj is the fraction of the set (Yi) corresponding to sleep stage j. The node is cut in half if the gain in information is positive. If it's not positive, the node will keep its current state and take on the characteristics of a leaf node with a specific label. The best information gathered from the residual attributes determines which ones to collect.

REPTree uses reduced error pruning to refine the decision tree further. It takes a separate validation dataset or uses cross-validation to estimate the error rate of each subtree. Pruning is performed by replacing each subtree with its majority class, reducing the complexity of the tree while maintaining its accuracy. One of the advantages of REPTree is its ability to handle both categorical and numerical features. It can handle missing values and binary and multi-class classification problems. However, like other decision tree algorithms, REPTree may be prone to overfitting if not properly pruned or regularized.

6.5.5.5 Sequential Minimal Optimization

Minimal sequential optimization (SMO) is a machine-learning classification technique for breaking quadratic problems into smaller problems that solve individually [247]. SMO avoids time-consuming quadratic optimization problems using an analytical approach. The support vector machine is trained and solved mathematically as follows (6.21):

$$\sum_{i=1}^{n} y_i \alpha_i = 0 \tag{6.21}$$

Where α_i, is Lagrange multiplayer and $y_i \epsilon$ {-1, +1} is the binary classification for the dataset. The SMO is an iterative algorithm for solving optimization problems by breaking the problem into serious problems. It is a technique for solving quadratic problems based on the support vector machine learning approach.

The new hybrid technique uses features and information using the clustering technique to the data set. The proposed scheme uses ten-fold cross-validation using 80% of training and 20% of testing using the benchmark dataset for attack detection and classification. The clustering technique divides the dataset into groups with various dimensions making new discrete features. The new clustering scheme uses the Gaussian mixture model (GMM) to develop new features [144], as shown in Figure 6. 10. The GMM method is a clustering model without any training requirement. After adding and allocating additional features, the dataset is modified for better performance and accuracy.

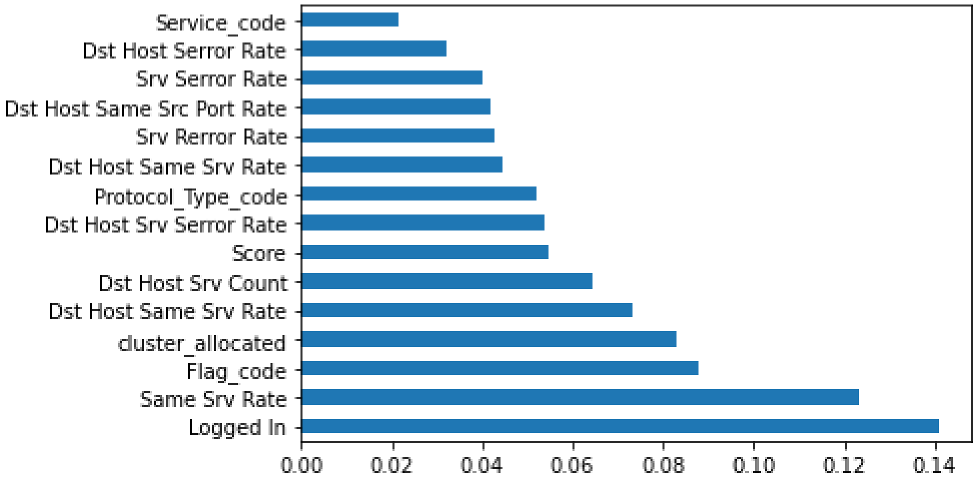


Figure 6. 10: Input extraction feature after modified NSL-KDD using Gaussian mixture clustering.

The proposed system uses training and testing procedures to detect and classify attacks using hybrid machine-learning techniques. The training model predicts the information of the nodes to classify as usual or malicious nodes using a benchmark dataset. The training process constructs a competitive learning model without external support for classifying attacks using the activated neurons to pass the output. Four machine learning models are utilized for the proposed system using hybrid supervised and unsupervised machine learning techniques, including logistic regression, support vector classifier, random forest, and Naïve Bayes using 80% for training and 20% for testing

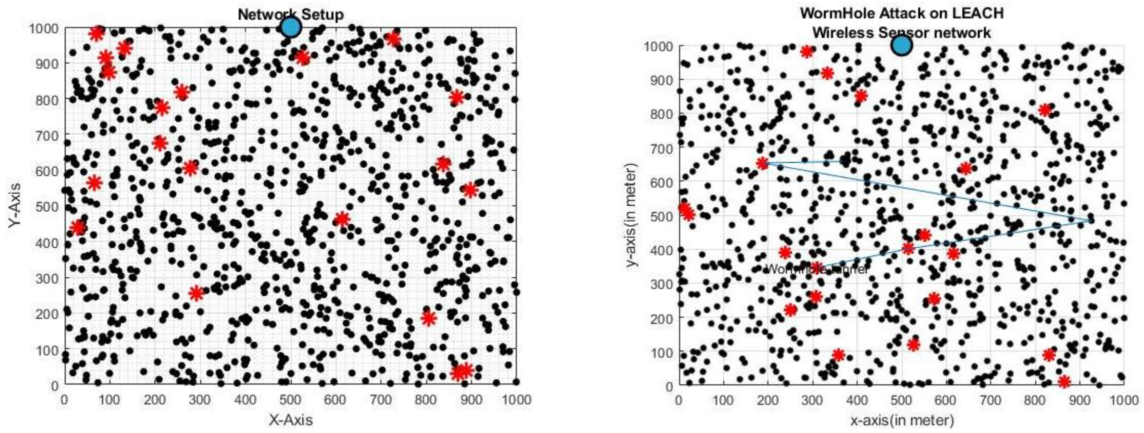
6.6 Simulation and Testing

The proposed system is designed to enhance the security and lifetime of the network using a secure data aggregation approach. MATLAB R2021a was used to plan the networks and execute the simulations on a computer with two Intel Xeon Silver 4214 CPUs running at 2.20GHz and 2.19GHz, with 128GB of installed RAM (126GB available), running Windows 10 64-bit on an x64-based processor. WEKA java Programming toolbox and python are used for data processing and analysis utilizing the machine learning classifiers. To simulate a WSN, a 1000 m x 1000 m monitoring area is populated with 1000 static sensor nodes placed at random. Keep the BS out of sight of the monitoring equipment by placing it at position (500,500). All sensor nodes are initially supplied with 0.5 J of energy. This means that the network's total starting energy is 50 J. The assumed size of a data packet to transmit sensed information is 4000 bits. The monitored region has been partitioned into 10 m 10 m granules. For clarity, we omit the control packet overhead while a path is established in the network. The network's energy usage was evaluated using the energy model given in Section 3. The simulation tests were run for one thousand WSN installations, and the average results from those runs were utilized to compile this comparative description of the protocols. The proposed hybrid fuzzy-based routing protocol is based on existing MATLAB code for implementing FD-LEACH, OCM-FCM, MH-EEBCDA, and similar protocols for detecting routing attacks in WSNs. Table 6. 2 shows the parameters of the simulation setup for network deployment.

Table 6. 2. Simulation setup for the proposed network model based on a wormhole attack scenario.

Parameter	values	Parameter	values
Maximum sensor nodes	1000	Number of attacks	20
Protocol type	Clustering	Sink Node position	1000, 500
Deployment Area	1000×1000 m^2	Initial Energy	0.5 J
Number of clusters	10	Data size	4000 Kbits
Radius	250	Attack type	Wormhole

The simulation results show that wormhole attacks create tunnels between two malicious nodes and increase the nodes' power consumption, making the network lifetime short. Figure 6.11 (a) and (b) show that the dynamic deployment and the routing wormhole illustrate how it causes packet dropping and sending malicious packets to the other nodes, respectively.



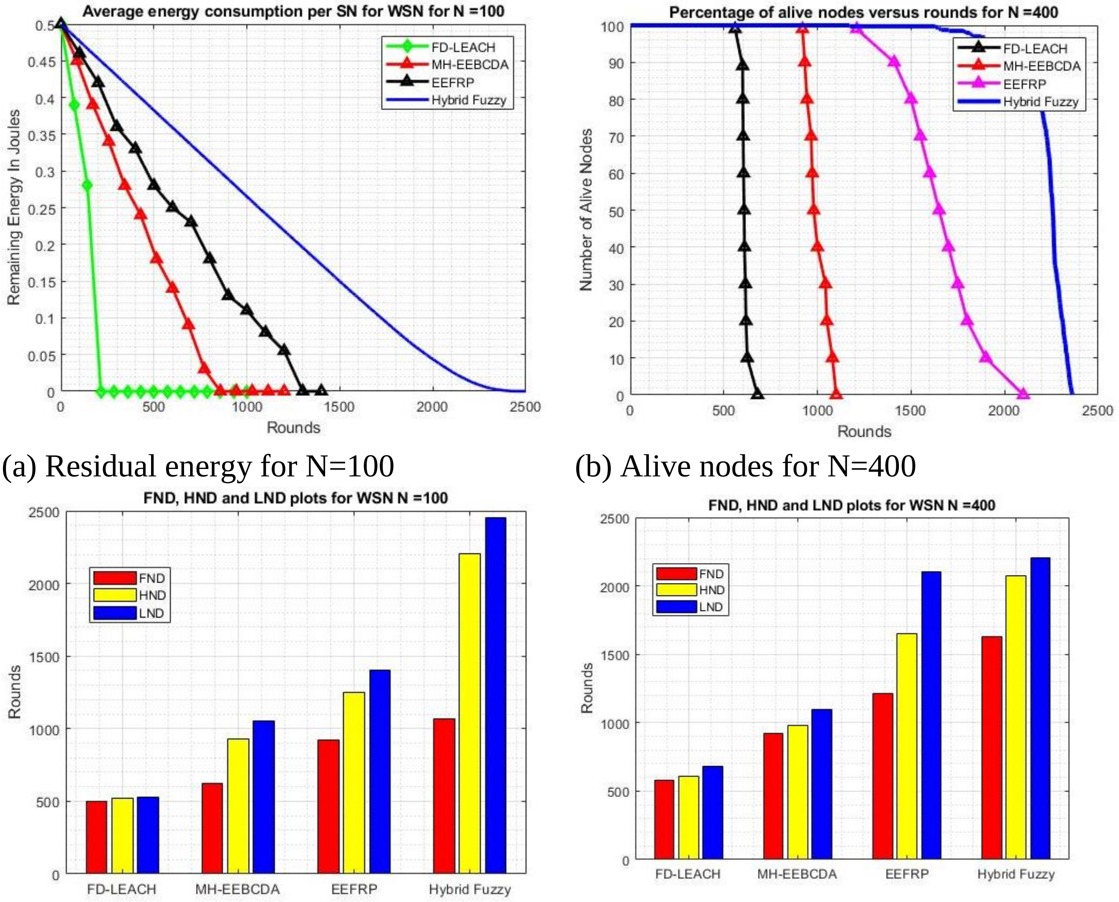
(a) WSN deployment (b) Routing discovery and wormhole tunnel

Figure 6.11. Wireless sensor network deployment and wormhole attacks exploit the routing information from the cluster heads to the base station.

By adjusting the number of nodes and the number of rounds used in each cluster head election using different clustering protocols, we may examine how well the suggested technique performs. The energy and security performance of the network improved by the hybrid fuzzy-based approach.

6.6.1 Energy Consumption Analysis

Data aggregation, transmission, and reception at each node during normal network operation contribute to the total energy consumption. Figure 6. 12 (a) and (b) are the scatterplot showing the relationship between the percentage of total energy consumed and alive nodes concerning the number of rounds for all of the tested procedures and the suggested protocol. The figures clearly show that the suggested protocol uses significantly less total energy than FD-LEACH (Fuzzy Decision based on Low Energy Low-Energy Adaptive Clustering Hierarchy), MH-EEBCDA (Multi-hop Energy Efficient and Balanced Cluster-Based Data Aggregation Algorithm), and OCM-FCM (Optimal Clustering Mechanism Fuzzy-C Means), and EEFRP (Energy Efficient Fuzzy Routing Protocol). The chart also shows that while the suggested protocol used up half of the network's energy, OCM-FCM and MH-EEBCDA used up 67%, and FD-network LEACHs became fully inoperable.



(a) Residual energy for N=100

(b) Alive nodes for N=400

(c) Comparative graph for N=100

(d) Comparative graph for N=400

Figure 6. 12. Comparative energy and network lifetime analysis using various secure clustering protocols and varying the number of sensor nodes.

This is because FD-LEACH uses probabilistic methods to choose CHs, leading to an uneven distribution of CHs in WSN and increased intra-cluster distances. As a result, it resulted in increased network energy consumption despite employing fuzzy sets and decisions during multi-hop route building. When a sensor node in a network's energy reserves runs out, it is removed from service as a viable part of the network. As the number of inactive nodes rises, the network's efficiency declines. As a result, the network's health is directly tied to the number of dead sensor nodes. Figure 6. 12 (c) and 8(d) show a scatter plot comparing the round times at which the Last Node becomes Dead (LND), Half Nodes become Dead (HND), and First Node becomes Dead (FND) occur across all protocols (d). The FND round is when the network is stable and operating. HND is a round number indicating how long the network takes to lose half its original capacity. A lifetime of Network Device (LND) is a round number that indicates the complete lifespan of the network. The figures also compare the suggested protocols (MH-EEBCDA, FD-LEACH, and OCM-FCM) using a scatter plot of FND, LND, and HND round numbers. The figures show that the proposed protocol's FND round is later than the FND rounds of FD-LEACH, OCM-FCM, and MH-EEBCDA by 563, 431, and 293, respectively. Additionally, the suggested hybrid fuzzy-based protocol has a round number for HND that is 25% better than OCM-FCM and 28% better than MH-EEBCDA.

When assessing a network's reliability, the trustworthiness of its base station is typically taken as a starting point [4]. It can monitor the CHs using several methods, including machine learning-based methods and anomaly detection methods because it has more resources at its disposal than the vast majority of sensor nodes. The BS uses the following metrics to assess the reliability of its CHs' communications, data, and energy infrastructures. The election phase guarantees the correct selection of CH. Malicious nodes are identified and removed from the network during the monitoring phase. We gave each node a certain amount of weight for the CH pick. Energy, physical distance, and the level of trust are all part of the weight. The level of trust was seen as crucial in choosing the CH. Cluster formation's validation step was implemented to fine-tune the network's total number of clusters. Our trust model for keeping tabs on the internet works by running a series of tests in rapid succession, focusing on a specific aspect of trust, such as the reliability of a network connection, the safety of stored data, or the reliability of a power grid. Each time we wished to verify the sensor nodes' reliability, we checked their link quality indicator (LQI) value to ensure we weren't unfairly excluding perfectly innocent nodes just because they had a weak wireless connection. Each cluster member keeps tabs on their CH's conduct and removes it from the group if it starts misbehaving.

6.6.2 Evaluation Metrics

The effectiveness of the proposed system is assessed using the metrics of the confusion matrix. These include accuracy, sensitivity, specificity, and training time. The new proposed approach is evaluated the evaluation metrics, including power consumption, detection rate, network lifetime, and detection accuracy of the attacks in the network. We also measure the efficacy of the suggested system to the following metrics: network scalability; events and communication overhead; communication range; communication failure; communication failure rate; aggregation ratio; and network load.

Network lifetime: The operational time the network performs the dedicated task. It can be computed when the source node energy drains to transfer to the base station. This shows that the loss of nodes leads to the loss of network functionality [93].

Confusion matrix: a confusion matrix creates a baseline for calculating the parameter metrics. The number of instances is tabulated as four values: true negatives, true positives, false positives, and false negatives. Table 6 shows the evaluation metrics for the proposed approach.

Accuracy (Acc.): It is computed by the number of packets lost during data transmission for a long period. Correctly detected attacks from the total malicious attacks. Mathematically it is computed as (6.22):

$$\text{Accuracy}=\frac{\text{TN+TP}}{\text{TN+TP+FN+FP}} \tag{6.22}$$

Detection Rate (DR): describes correctly Classified attacks from the total malicious nodes [5], [196]. Recall and precision are also performance metrics for classifying attacks against the dataset. Precision describes correctly retrieved attacks from the total number of instances, whereas recall details the binary classification analogous to positive predicted values. This can be expressed as in equation (6.23):

$$\text{Detection Rate}=\frac{\text{TP}}{\text{FP+TP}} \tag{6.23}$$

$$\text{Recall}=\frac{\text{TP}}{\text{FP+TP}}$$

$$\text{Precision}=\frac{\text{TP}}{\text{FP+TP}}$$

Sensitivity (Recall or True Positive Rate): Sensitivity measures the system's ability to correctly identify positive instances (true positives) out of all positive ones. It is calculated as the ratio of true positives to the sum of true positives and false negatives. Sensitivity is particularly important when minimizing false negatives or detecting rare events.

Specificity: We also utilized specificity and false positive rate for the number of negatives and misclassification of attacks. Specificity is the number of negatives correctly identified from the total sample traffic. False-positive rate describes the incorrect classification of legitimate nodes as an attack and is computed by (6.24).

$$\text{Specificity}=\frac{\text{TN}}{\text{FN+TP}} \tag{6.24}$$

$$\text{False positive rate}=\frac{\text{FP}}{\text{FP+TN}}$$

F-Measure: The weighted harmonic mean of recall and precision of the test. The MCC is also utilized for measuring the imbalance of the dataset for performance measurement. The F-Measure can be formulated by the equation (6.25) as shown below:

$$\text{F-Measure}=\frac{2\times\text{R}\times\text{P}}{\left(\text{R+P}\right)} \tag{6.25}$$

The other performance metrics are shown below in Table 6. 3.

Table 6. 3. Performance metrics and technical description.

Measure	Technical Description
ROC	It is the threshold curve for the classification model plotted using TPR and FPR.
Kappa	It checks the reliability of the classification technique [244]. It is calculated by assessing the agreement between the model's predictions and the ground truth.
MAE	MAE error is the average absolute difference between projected and actual values. It measures model accuracy by comparing two continuous variables.
Time	WSN security depends on time. It synchronizes clocks, and prevents replay attacks..
Energy	WSN security research focuses on energy. WSN security depends on energy efficiency and design. Encryption, authentication, authorization, and access control use energy.

6.6.3 Intrusion Detection Analysis

The system performance for the model is measured using the NSL-KDD Dataset as a benchmark with machine learning classifiers. At this time, two datasets, specifically NSL-KDD and UNSW-NB15, are being utilized to identify the assaults that are being made against the system [248]. In this section, the test configurations that need to be selected to validate the presence of the proposed framework based on hybrid machine learning are presented. In addition to this, the hybrid IDS is correlated with several critically important parameters. The evaluation of the suggested method is based on separating a denial of service attack from a normal network to perform multiclass classification. Whether the classification was done correctly or not, the results were expressed as True positives, True negatives, False positives, and False negatives. Table 6. 4 shows the classification evaluation for different classes of attacks. The system's effectiveness is evaluated using the detection accuracy of attacks and the system's processing speed. The results show that random forest achieves better attack detection accuracy for the heterogeneous dataset.

Table 6. 4. The performance evaluation of the proposed system using machine learning classifiers.

Metrics	REPTree	NB	RF	SMO	J48	Bagging
Accuracy	99.67%	84.60%	99.91	98.32	99.76%	99.69%
Error	0.3302%	15.40%	0.0905%	1.68%	0.2374%	0.3056%
Kappa	0.9942	0.7497	0.9984	0.97	0.996	0.9947
MAE	0.002	0.0613	0.0015	0.24	0.0012	0.0021
RMSE	0.0352	0.2426	0.02	0.32	0.0295	0.0306
RAE	0.8766%	26.7746%	0.6467%	105.08%	0.5292%	0.9042
RRSE	10.4024%	71.6966%	5.9068%	93.52%	8.7087%	9.0326
TTBM	3.92	0.77	50.82	324	16.51	29.42

The proposed system's ability to detect and classify denial of service (DoS) assaults in WSNs was evaluated based on the metrics presented in Table 6. 5, generated using the NSL-KDD dataset. In the process of making decisions, the random forest is a useful method for both the detection of attacks and the classification of diverse datasets.

Table 6. 5. Detection and Classification of the system using random forest classifier for the five classes of attacks in the Dataset.

Attacks	F1score	MCC	TPR	FPR	Precision	Recall	ROC	PRC
Normal	0.999	0.998	1.000	0.001	0.999	1.000	1.000	1.000
DoS	1.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000

R2L	0.981	0.981	0.968	0.000	0.994	0.968	0.999	0.997
Probes	0.998	0.998	0.997	0.000	0.999	0.997	1.000	1.000
U2R	0.733	0.742	0.635	0.000	0.868	0.635	0.990	0.811

Table 6. 6 compares the relative performance of various machine learning classifiers when it comes to the detection and classification of attacks. Random forests have a higher average detection rate, which allows for more accurate detection of attacks in datasets with varying degrees of homogeneity in wireless sensor networks. The proposed technique achieved better attack detection accuracy of 100%, 100%, 96.8%, 99.7%, and 63.5% for normal, DoS, R2L, Probes, and U2R, respectively, using the benchmark NSL-KDD dataset using 80% training samples and 20% testing samples. Superior quality of service (QoS) factors like energy, longevity, and security can be attained by combining the suggested technique's secure clustering and data aggregation with an intrusion detection technique based on machine learning [249]. The proposed model chooses initial tentative cluster heads (TCHs) based on three input parameters: residual energy, distance to BS, cost, number of connections, and distance to neighbours.

Table 6. 6. Detection rate performance of different classifiers using 10-fold cross-validation for various classes of attacks.

	Classes of Attacks				
Classifiers	Normal	DoS	R2L	Probes	U2R
Random Forest	1.000	1.000	0.968	0.997	0.635
REPTree	0.999	0.999	0.958	0.981	0.385
Naïve Bayes	0.801	0.949	0.448	0.732	0.904
SMO	0.989	0.983	0.779	0.969	0.404
J48 Tree	0.998	0.999	0.946	0.993	0.538
Bagging	0.999	1.000	0.952	0.981	0.442

The comparison of six machine learning classifiers is shown in Table 6. 7 for the detection and Classification of DoS attacks classes using the NSL-KDD dataset as a benchmark. This indicates that the random forest classifier effectively detects attacks in aggregated data heterogeneous datasets.

Table 6. 7. Comparison of performance of different machine learning classifiers for detecting the class of attacks.

Classifiers	TPR	F-M	FPR	Recall	Precision	MCC	ROC	PRCA.
RF	0.999	0.999	0.001	0.999	0.999	0.999	1.000	1.000
REPTree	0.997	0.997	0.002	0.997	0.997	0.995	0.999	0.998
NB	0.846	0.878	0.044	0.846	0.917	0.801	0.972	0.946
SMO	0.983	0.983	0.014	0.983	0.983	0.971	0.990	0.976

J48 Tree	0.998	0.998	0.002	0.998	0.998	0.996	0.999	0.997
Bagging	0.997	0.997	0.002	0.997	0.997	0.995	1.000	0.999

Evaluated malicious activity detection rate using secure data aggregation protocol achieves 98.84% malicious node detection. The proposed scheme is more effective, with a detection accuracy of 99.91% using the random forest classifier. The analysis and effectiveness of the new approach are validated and confirmed by comparing it with previous research work for attack detection in WSNs. Table 6. 8 compares the various models using the benchmark dataset.

Table 6. 8. The comparative performance of the summary for each classification Model using CIC-IDSS2017.

Metrics	RF	MLP	Bagging	AdaBM	J48	REPT
CC	99.83%	98.184%	99.82%	94.61%	99.82%	99.80%
ICC	0.171%	1.82%	0.1832%	5.38%	0.18%	0.1961%
Kappa	0.9964	0.9609	0.9961	0.8833	0.9962	0.9958
MAE	0.0007	0.0081	0.0008	0.0667	0.0007	0.0008
RAE	0.4423%	5.1817%	0.4849%	42.4763%	0.4574%	0.4972%
RMSE	0.0193	0.0726	0.0197	0.137	0.0204	0.0211
RRSE	6.8817%	25.9023%	7.0443%	48.9029%	7.2724%	7.5322%
TTBM	323.75	576.07	227.2	32.12	66.07	26.39

The random forest classifier has greater detection accuracy with the same cross-validation, 10 and 80% of training, and 20% of testing the dataset to detect the class of attacks. The performance metrics of the random forest classifier are shown in Table 6. 9.

Table 6. 9. Detailed detection accuracy of the class of attacks based on random forest CIC-using IDSS2017 dataset.

	Performance Metrics (%)							
Attacks	TPR	FPR	PR	RC	FScore	MCC	ROC	PRC
Normal	0.999	0.002	0.999	0.999	0.999	0.996	1.000	1.000
Slowloris	0.795	0.001	0.772	0.795	0.783	0.782	0.999	0.889
Slowhttptest	0.996	0.000	0.997	0.996	0.997	0.997	0.999	0.999
Hulk	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000
GoldenEye	0.998	0.000	0.999	0.998	0.999	0.999	1.000	1.000
Heartbleed	0.909	0.000	1.000	0.909	0.952	0.953	1.000	0.986
Wgt. Avg	0.998	0.001	0.998	0.998	0.998	0.996	0.998	0.997

The effectiveness of the random forest technique is compared with other machine learning approaches for its validation, as shown below in Table 6. 10. The actual positive metrics are used for measuring the detection accuracy of the different classes of attacks. The average detection accuracy of the three machine learning techniques is 99.83%, 98.20%, 99.20%, and 94.61% for the random forest, multilayer perception, bagging, and AdaBoost.M1, respectively. It is also compared with J48 and REPTree machine learning schemes for flooding attack classification.

Table 6. 10. Performance comparison of the true positive rate for the different classes of attacks using machine learning schemes.

	Machine learning techniques					
classifiers	RF	MLP	Bagging	AdaBoost.M1	J48	REPTree
Normal	0.999	0.998	0.998	0.972	0.998	0.998
Slowloris	0.795	0.096	0.828	0.000	0.890	0.839
Slowhttptest	0.996	0.000	0.991	0.000	0.990	0.991
Hulk	1.000	0.972	1.000	0.963	1.000	1.000
GoldenEye	0.998	0.952	0.996	0.000	0.996	0.995
Heartbleed	0.909	0.000	0.909	0.000	0.909	0.909

The performance of the proposed system is compared with M. Agarwal et al.[250] using machine learning schemes. The detection rate and accuracy of the proposed scheme random forest technique is effective flooding attack detection.

The proposed scheme achieves better attack detection accuracy of 100% using a mixed random forest. The classification technique is more effective than M. N. A. Shaon and K. Ferens [187] using the artificial neural network (ANN) approach with a detection rate of 99.72% for the uniform distribution of nodes. The performance of the proposed system is evaluated using the confusion matrix as shown in Table 6. 11 and the table for the standard classes in the dataset labeled as binary 0 for both standard and hybrid machine learning classifiers.

Table 6. 11. Performance comparison of the classifiers for regular classes.

	Machine learning models				Hybrid machine learning			
Classifier	PR	RC	F-M	AUC	PR	RC	F-M	AUC
LR	0.97	0.96	0.96	0.991	0.98	0.96	0.97	0.991
SVM	1.00	0.99	0.99	0.999	1.00	0.99	0.99	0.999
RF	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.000
NB	0.86	0.94	0.90	0.963	0.87	0.94	0.90	0.962

The comparison performance for classification and detection of the class of attacks suggests that the random forest classifier archives a better detection rate of 100%. Table 6. 12 shows the performance comparison of the evaluation metrics for standard and hybrid machine learning classifiers.

Table 6. 12. Performance comparison of the classifiers for attack classes.

	Machine learning models				Hybrid machine learning			
Classifier	PR	RC	F-M	AUC	PR	RC	F-M	AUC
LR	0.95	0.97	0.96	0.991	0.96	0.97	0.96	0.992
SVM	0.98	1.00	0.99	0.999	0.99	1.00	0.99	0.999
RF	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.000
NB	0.92	0.83	0.87	0.963	0.93	0.84	0.88	0.963

The support vector classifier's execution time is much greater than the others in standard and hybrid schemes. Generally, the time taken for the hybrid technique is less than that of the standard machine learning techniques, as shown in Table 6. 13. The random forest classifier is more effective for detecting and classifying attacks in standard and hybrid schemes.

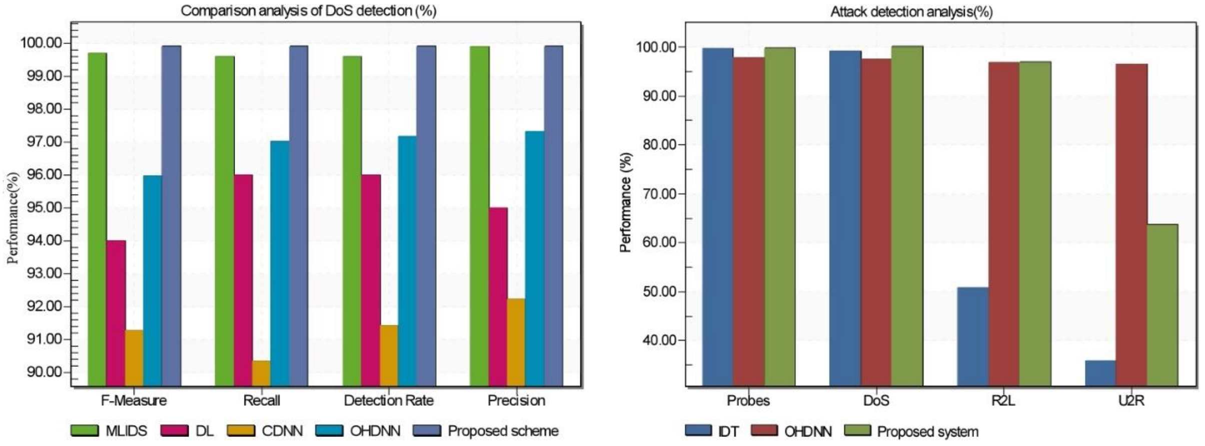
Table 6. 13. Testing accuracy and time duration for classification of attacks.

	Normal		Hybrid	
Classifier	Accuracy	Time	Accuracy	Time
Logistic Regression	0.96	0.12146	0.97	0.12923
Support Vector	0.99	11.05374	0.99	9.80267
Random Forest	1.00	0.92058	1.00	0.87550
Naïve Bayes	0.89	0.01562	0.89	0.0

The various comparison performance with previous works shows that the proposed system effectively detects attacks in WSNs.

6.6.4 Analysis of Comparison Performance

The error rate of the proposed system is low compared to [249] developed for intrusion detection and quality of service enhancement in WSNs using a secure clustering technique. The sachem utilized NSL-KDD 2015 and CICIDS 2017 datasets for attack detection and analysis with error rates of 0.04 and 0.05, which are large values compared to the proposed approach with an error rate of 0.02 using the random forest classifier. M. Anbarasan et al.[134] Presented the detection of flooding disasters on IoT using a convolutional neural network (CDNN) with normalization and imputation functions for data processing. Figure 6.13 (a) shows the validation of the proposed system with CDNN using detection rate, F-measure, recall, precision, and accuracy performance metrics. S. M. Kumar [46] proposed a feature selection approach based on feature selection to pick the most contributing features, and an optimized hybrid deep neural network (OHDNN) is provided for classification. Convolutional neural networks (CNNs) and long short-term memories (LSTMs) have been combined to create the hybrid deep neural network (LSTM).



(a) Comparison of performance. (b) Analysis of attack detection comparison.

Figure 6.13. Performance comparison of the proposed system with various recent works using performance metrics.

T. Rose et al. [53] Proposed hybrid k-means and support vector machine learning for detecting malicious attacks for the internet of energy using the KDD-CUP'99 dataset, achieving 99.9% average detection accuracy less than the random forest approach. U. Ghugar et al. [5] Presented the LB-IDS method for detecting sinkhole, cross-layer, back-off manipulation, and jamming attacks with a detection accuracy of 89.83%, 92.16%, 95%, and 96.83%, respectively. This confirms that the proposed system effectively detects multiple attacks for a heterogeneous dataset containing classes of attacks. Figure 6.13 (b) shows the comparison of the proposed system with F. Y. Yavuz et al. [66] for the detection of routing attacks using the deep learning (DL) technique. The detection accuracy of the proposed system is compared with the intelligent decision tree (IDT) classification presented by P. Nancy et al. [155] for the detection of the four classes of attacks, as shown in Figure 6.13 (b). The average detection accuracy of the random forest classifier is effective and performs better for detecting Probes, U2R, R2L, and DoS attacks in wireless sensor networks using benchmark datasets.

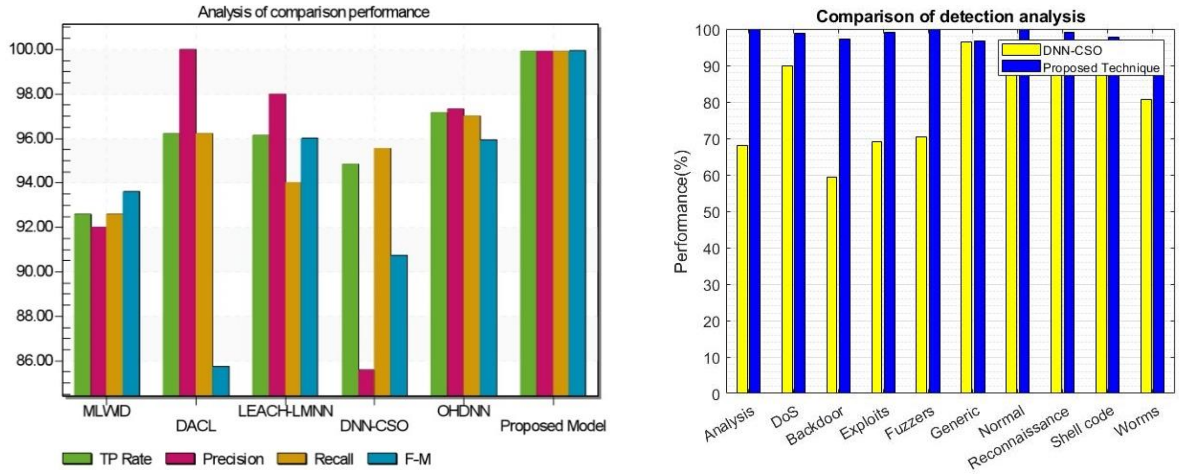
The effectiveness of the new approach technique is further compared with N. N. Gana and S. M. Abdulhamid [251] related work for detecting phishing attacks and achieves average detection accuracy of 98.38% using 10-fold cross-validation using a random forest classifier. Table 6. 14 compares the machine learning technique base for wireless intrusion detection (MLWID) using the wireless intrusion detection dataset as a benchmark conducted by M. Nivaashini and P. Thangaraj [196]. Table 6. 14 compares the random forest technique with J.

Ramprasath and V. Seethalakshmi [252] examined flooding attack detection using a dynamic access control list (DACL). M. Mittal et al. [188] proposed efficient and reliable protocols using machine learning for intrusion detection systems (IDS) With an average detection accuracy of 96.15%, including two well-known energy efficient protocols: Low-Energy Adaptive Clustering hierarchy (LEACH) and a Levenberg-Marquardt neural network (LEACH -LMNN).

Table 6. 14. Performance comparison of the proposed system with recent works using various metrics of detection and classification.

Ref.	Technique	TP Rate	Precision	Recall	F-M	FPR
[251]	MLWID	92.6	92	92.6	93.6	8.5
[252]	DACL	96.2	100	96.2	85.7	0.002
[188]	LEACH-LMNN	96.15	98.00	94.00	96.00	0.05
[210]	DNN-CSO	94.85	85.59	95.53	90.72	0.045
[46]	OHDNN	97.17	97.32	97.02	95.92	4.8
-	Proposed Model	99.9	99.9	99.9	99.9	0.001

R. Khilar et al. [210] proposed a deep neural network using a chicken swarm optimization (DNN-CSO) algorithm-based anomaly detection model for an Internet of Things (IoT) network by utilizing the UNSW-NB15 dataset as a reference. They achieved average detection accuracy of 96.53%, as shown in Figure 6.14 (a) and (b). The DNN has already demonstrated effectiveness in a variety of domains where it can be put to use.



(a) Attack detection Model comparison. (b) Comparison for each attack.

Figure 6.14. Performance comparison of recent works and various classes of attacks.

M. F. Suleiman and B. Issac [253] evaluated the performance of machine learning classifiers-based intrusion detection (MLWIDS) using NSL-KDD and USNW-NB15 datasets for detecting malicious attacks in corporate networks with a detection accuracy of 99.76% and 90.14%, respectively. The classifiers are evaluated concerning testing time, detection accuracy, and false-positive rate. This proves that the new approach achieves better attack detection and classification detection using a benchmark dataset with a detection accuracy of 99.91%. The performance of the proposed scheme is further validated by comparing it with other previous works. N. A. Awad [254] presented and achieved an average detection accuracy of 99.2% using decision with KDD99 as a benchmark for attack detection and classification that is less than the proposed approach having a detection accuracy of 99.91%.

Machine learning was investigated for intrusion detection in secure, clustered, IoT-based WSNs with restricted resources. We created an effective IDS to detect suspicious behaviour on unsecured IoT networks by expertly combining feature dimensionality reduction and machine learning. We tested our strategy for ML-based IDS on the UNSW-NB15 and NSL-KDD datasets. Security concerns have slowed the Internet of Things growth. Machine learning-based IDS may monitor security.

6.7 Conclusion

Secure data aggregation and clustering improve the lifetime and security performance of the network. The aggregation node collects and forwards the data to the sink node via the cluster head from the sensor nodes monitored by the central node. The proposed hybrid fuzzy-based routing protocol is effective for throughput, energy efficiency, network lifetime, and overall network security performance. The base station verifies the aggregated data to detect malicious routing attacks using security protocols. This approach, clustering, and aggregation enhance the network lifetime and security by applying machine learning classifiers for attack detection and classification in WSNs. The effectiveness of the proposed system is evaluated using machine learning classifiers with the NSL-KDD Dataset as a benchmark. The random forest achieves a better version of attack detection and classification with an average detection accuracy of 99.91% in hierarchical wireless sensor networks. Random forest techniques are effective for a heterogeneous dataset for attack classification and detection in WSNs. The comparison performance confirms that the proposed system effectively detects DoS attacks in WSNs using effective network planning and design with a benchmark dataset for evaluation.

Future works will investigate secure data aggregation techniques using hybrid and advanced machine learning-based optimization in wireless sensor network planning and configuration approaches. We will explore another heterogeneous dataset using machine learning classifiers for secure data aggregation and attack detection.

Chapter 7

7 CONCLUSION AND FUTURE WORK

In this section, we will briefly discuss the main results and findings of the current study. No research is ever considered complete unless it points to several avenues for further study, no matter how substantial its contributions are. So, in Sections 7.1 and 7.2, we briefly address the significant contributions, limitations and directions to future studies that can be explored in further depth.

7.1 Major contributions

This study uses various security schemes to enhance the security of wireless sensor networks (WSNs) integrated with the Internet of Things (IoT). The thesis focused on identifying the most effective security schemes for protecting the data and communications in WSNs connected with IoT. The study also investigated the security challenges related to the architecture of integrating WSNs with IoT and identified possible solutions. Furthermore, we evaluated the performance of the security schemes in terms of resilience and scalability. Finally, we addressed and developed several security frameworks for WSNs integrated with IoT based on different techniques summarized below:

- Advanced intrusion detection systems based on machine learning techniques effectively detect and classify attacks for scalable and manageable in hierarchically distributed wireless sensor networks. This research aims to create a classification model for an advanced intrusion detection system based on hybrid machine learning, specifically tailored for use in wireless sensor networks to detect intrusions. Each sensor node collects information on the state of its features and reports it to the cluster's central processing node. The cluster leader checks the data and then forwards it to the main cluster head. The proposed hybrid machine learning models use training and testing data to identify attacks. Our suggested IDS-HML outperforms state-of-the-art systems regarding detection and localization accuracy in a simulated attack on a WSN. Comparing the hypothetical outcomes to earlier research shows that they are credible. The simulation results show that the proposed system is effective for detecting routing attacks with a localization accuracy of 99.46% of the wormhole routing attacks. The effectiveness of the suggested system has been measured in accuracy, precision, TP Rate, FP Rate, F-Measure, Mean squared error, and Time. The designed IDS-HML achieved 99.82%, 99.91%, 99.85%, 99.82%, and 100% for average detection accuracy, precision, F1-score, recall, and CLK-Means respectively, in the presence of normal and intrusion traffic using CICIDS2017 dataset as a benchmark for multiclass and binary classifications. This work is implemented using MATLAB for network planning and simulation of attack scenarios. The Python libraries are utilized for hybrid machine-learning classification techniques. This model uses logic rules for decision-making and interpretable predictive models.

- To identify and locate multiple assaults in WSNs, we propose a multilayer perceptron artificial neural network (MLPANN) in this work. Using the UNSW-NB, WSN-DS, NSL-KDD, and CICIDS2018 benchmark datasets, the suggested technique had an average detection accuracy of 100%, 99.65%, 98.95%, and 99.83 % for the various malicious nodes. With an average localization accuracy of 99.12% employing 160 beacon nodes, the improved localization approach is more successful and performs considerably by 20% than the distance vector hop technique. Preliminary studies employing the ANN classification methodology confirm the validity of the suggested method. The suggested system's detection and localization accuracy for various assaults is measured using the datasets. The proposed method's effectiveness is measured using various metrics, including detection rate, ROC, false-positive rate, network lifetime, residual energy, and area under the curve. Target fields were simulated using a hierarchical combination of beacon, sensor, and malicious nodes. It is suggested to improve the accuracy of identification and localization of malicious nodes using various methods in WSNs. As we progress, we'll add new types and techniques of attack to the mix. The findings demonstrate that the security performance of the proposed system is effective for large-scale, scalable wireless sensor networks that contain heterogeneous and homogeneous sensors.
- The suggested approach uses a blockchain-based security mechanism to identify and pinpoint rogue nodes in hierarchically distributed IoT-WSNs. Blockchain technology can detect malicious nodes in IoT-based wireless sensor networks by using XGBoost and CLK-Means machine learning federated classifiers for multiclass and binary classification approaches. This method, which employed feature assessment and cascade encryption, aimed to both increase network throughput and ensure the secure delivery of services in the event of an attack. Simulation and classification results show that the proposed system is

adequate for malicious node detection and localization, with detection accuracies of 99.95% and 100% for XGBoost and CLK-Means, respectively, based on the multiclass and binary classification approach with the CICIDS2017 benchmark dataset. Hybrid federated machine learning is a new method for detecting and localizing assaults in IoT-WSNs. Results show random forest outperforms other approaches in classifying complex network traffic data and identifying malicious nodes. To evaluate how well the suggested paradigm works in providing services in a secure and timely manner, we additionally track metrics like node honesty, end-to-end delay, and transaction latency. There needs to be further study into detecting the position of malicious nodes because sensor nodes are often deployed in unsupervised settings where they are vulnerable to various routing attacks.

- Secure data aggregation and clustering improve the lifetime and security performance of the network. The aggregation node collects and forwards the data to the sink node via the cluster head from the sensor nodes monitored by the central node. The proposed hybrid fuzzy-based routing protocol is effective for throughput, energy efficiency, network lifetime, and overall network security performance. The base station aggregates verification to detect malicious routing attacks using security protocols. This approach, clustering, and aggregation enhance the network lifetime and security by applying machine learning classifiers for attack detection and classification in WSNs. The effectiveness of the proposed system is evaluated using machine learning classifiers with the NSL-KDD Dataset as a benchmark. The random forest achieves a better version of attack detection and classification with an average detection accuracy of 99.91% in hierarchical wireless sensor networks. Random forest techniques are effective for a heterogeneous dataset for attack classification and detection in WSNs. The comparison performance confirms that the proposed system effectively detects DoS attacks in WSNs using effective network planning and design with a benchmark dataset for evaluation.
- The proposed system uses intelligent network planning and deployment to optimize routing and clustering to enhance network performance and lifetime. The GA and PSO techniques utilized objective functions to find optimal solutions and improve network performance. The PSO-based multi-hop transmission effectively maximizes the network lifetime for a given number of iterations with a minimum routing cost. The hybrid GA-PSO enhances the network lifetime by 8.71% using the multi-hop transmission routing approach. The proposed system also uses a benchmark dataset using Logistic regression, support vector,

random forest, and Naïve Bayes machine learning models to detect attacks based on the standard and malicious activity of the nodes in the network. Hybrid machine learning classifiers detect different attacks using the benchmark dataset. The Gaussian mixture adds new features to the benchmark dataset by k-means clustering and principal component analysis for training and testing. The hybrid random forest archives average detection accuracy of 100% by training the modified dataset. The execution time and performance of the system are improved after the new feature are added to the benchmark dataset.

7.2 Limitations

The research thesis outlined focuses on enhancing the security of wireless sensor networks (WSNs) integrated into the Internet of Things (IoT) by employing various security techniques and methodologies. While the proposed research has promising objectives, it's essential to acknowledge certain limitations that may affect its scope and applicability. These limitations include:

Resource Constraints: Many IoT devices and sensors, especially those in WSNs, are resource-constrained in terms of processing power, memory, and energy supply. Implementing complex security mechanisms might impose additional computational and energy overhead, potentially affecting the overall performance and longevity of the devices.

Compatibility: Integrating security measures into existing WSNs and IoT ecosystems may require updates, patches, or hardware upgrades, which can be challenging and costly. Compatibility issues with legacy devices and systems might arise, limiting the deployment of enhanced security measures.

Scalability: As the IoT grows, scalability becomes a significant concern. Ensuring security solutions remain effective and efficient as the network scales up can be challenging.

Complexity: Implementing machine learning-based intrusion detection systems and intelligent routing methods can introduce network management and troubleshooting complexity. Users and administrators may require specialized knowledge to operate and maintain such systems effectively.

Data Privacy: The collection and analysis of data for security purposes raise concerns about data privacy and compliance with relevant regulations. Balancing the need for security with the privacy rights of individuals and organizations is a complex issue.

Benchmark Dataset Limitations: The effectiveness of machine learning-based security approaches heavily relies on the quality and representativeness of benchmark datasets. Limited or biased datasets may affect the accuracy and generalizability of the intrusion detection systems.

Adaptation to Evolving Threats: Security threats in the IoT landscape are continually evolving. The research may need to address the challenge of adapting security measures to new and emerging threats in real time.

Energy Efficiency: Energy-efficient security mechanisms are crucial in WSNs, where battery-powered devices are prevalent. Striking a balance between security and energy consumption is a critical consideration.

Cost: Implementing advanced security measures, especially machine learning ones, may require significant financial investments in hardware, software, and personnel training.

Network Heterogeneity: The IoT comprises various devices with varying capabilities and communication protocols. Ensuring that security schemes are compatible with this heterogeneous environment is complex.

Real-World Deployment: Implementing and testing these security measures in real-world IoT environments can be difficult. The research may need to rely on simulations or limited-scale testbeds, which may not fully capture the complexities of real-world IoT deployments.

Blockchain Overhead: While blockchain technology can enhance security, it also introduces overhead in terms of storage, computation, and communication. Balancing the benefits of blockchain with its associated costs is a challenge.

Algorithm Selection: The thesis mentions using machine learning methods against benchmark datasets, but selecting the right algorithms and training data can significantly impact the effectiveness of intrusion detection systems. The choice of algorithms and datasets must be carefully considered.

Evaluation Metrics: Properly evaluating the performance of security solutions is essential. Defining appropriate metrics for assessing the effectiveness, efficiency, and security of the proposed methods can be challenging.

Integration Challenges: Integrating various security components (intrusion detection, localization, blockchain, machine learning, etc.) into a cohesive system can be complex. Ensuring compatibility and interoperability among these components is crucial.

Adversarial Attacks: Adversaries may adapt to the proposed security measures and find new ways to compromise the network. Continuously updating and improving the security mechanisms to defend against evolving threats is a constant challenge.

Interoperability with Existing IoT Devices: IoT ecosystems consist of various devices from different manufacturers. Ensuring that the proposed security measures can be integrated with existing IoT devices and protocols may be challenging.

Addressing these limitations will be crucial to the success of our research thesis. Moreover, acknowledging these challenges and considering potential mitigations in our research plan will demonstrate a comprehensive understanding of the complexities in enhancing IoT-integrated WSNs' security.

7.3 Future works

This thesis contributes significantly to the field by thoroughly examining and analyzing various security methods and frameworks for IoT-WSNS. We explored the various security schemes in wireless sensor networks integrated into the Internet of Things. The existence of numerous devices, the dispersed structure of the networks, and the requirement for pervasive security all contribute to making the security demands in such networks greater than in conventional networks. Yet, the current findings raise several research topics worthy of further investigation. Future research should focus on developing a more secure framework based on various techniques to further improve the security of wireless sensor networks integrated into the Internet of Things.

Additionally, research should focus on developing methods and frameworks to evaluate the security of these networks realistically. This would allow researchers to understand such networks' threats and vulnerabilities better and design more secure systems. Several factors are enumerated as suggestions for further research:

- Although the proposed method performs well, it is essential to note that IoT-based WSNs are still susceptible to sinkholes, Sybil attacks, blackhole attacks, selective forwarding, and classification attacks not addressed in this study. The countermeasures module is only provided in concept, which is another shortcoming. Therefore, we plan to investigate and eventually offer specialized advanced hybrid intrusion detection systems for each type of assault utilizing benchmark datasets to evaluate hybrid machine learning techniques. In future work, we will explore collaborative advanced intrusion detection systems based on machine learning in IoT-based wireless sensor networks for different applications using benchmark datasets for evaluations.

- Developing robust defense mechanisms against adversarial attacks is crucial. Adversarial attacks involve introducing carefully crafted input data to deceive neural networks and cause them to make incorrect predictions. Research can focus on developing techniques to detect and mitigate such attacks. Investigating defenses against data poisoning attacks is important. These attacks involve maliciously manipulating the legitimate node to introduce vulnerabilities into the network. Developing methods to detect and mitigate data poisoning attacks can enhance the security of the network.
- For future large-scale deployments of secure, intelligent IoT-WSNs, we plan to research and create state-of-the-art blockchain-based secure IoT-WSNs using hybrid federated and machine learning approaches. Our research will focus on blockchain-based advanced hybrid access control techniques for locating and identifying rogue nodes in IoT-WSNs. We'll also use several benchmark datasets and hybrid routing protocols to create state-of-the-art multiclass and binary classification evaluation metrics.
- In the future, researchers will look at optimizing the setup and operation of wireless sensor networks by using hybrid and cutting-edge machine-learning techniques for gathering and securing large amounts of data. To secure data aggregation and detect attacks, we will also investigate another heterogeneous dataset using machine learning classifiers.
- To further improve security and performance in wireless sensor networks, researchers want to look into security hybrid methodologies with various datasets in the future. Hybridization brings useful features that boost the safety and efficiency of wireless sensor networks. They are also essential for methods of deployment and preparation that employ various routing algorithms.
- We will look into and implement alternative routing approaches for network planning and deployment and attach detection methods with a new benchmark dataset. In addition, alternative routing assaults with simulated detection of DoS attacks using different techniques while maintaining optimal efficiency and speed.

REFERENCE

[1] S. Pundir, M. Wazid, D. P. Singh, A. K. Das, J. J. P. C. Rodrigues, and Y. Park, "Intrusion Detection Protocols in Wireless Sensor Networks Integrated to Internet of Things Deployment: Survey and Future Challenges," *IEEE Access*, vol. 8, pp. 3343–3363, 2020, doi: 10.1109/ACCESS.2019.2962829.

[2] W. Zhou, P. Li, Q. J. Wang, and N. Nabipour, "Research on data transmission of wireless sensor networks based on symmetric key algorithm," *Meas. J. Int. Meas. Confed.*, vol. 153, p. 107454, 2020, doi: 10.1016/j.measurement.2019.107454.

[3] S. Dong, X. gang Zhang, and W. gang Zhou, "A Security Localization Algorithm Based on DV-Hop Against Sybil Attack in Wireless Sensor Networks," *J. Electr. Eng. Technol.*, vol. 15, no. 2, pp. 919–926, 2020, doi: 10.1007/s42835-020-00361-5.

[4] A. Saidi, K. Benahmed, and N. Seddiki, "Secure cluster head election algorithm and misbehavior detection approach based on trust management technique for clustered wireless sensor networks," *Ad Hoc Networks*, vol. 106, 2020, doi: 10.1016/j.adhoc.2020.102215.

[5] U. Ghugar, J. Pradhan, S. K. Bhoi, and R. R. Sahoo, "LB-IDS: Securing Wireless Sensor Network Using Protocol Layer Trust-Based Intrusion Detection System," *J. Comput. Networks Commun.*, vol. 2019, 2019, doi: 10.1155/2019/2054298.

[6] L. Han, M. Zhou, W. Jia, Z. Dalil, and X. Xu, "Intrusion detection model of wireless sensor networks based on game theory and an autoregressive model," *Inf. Sci. (Ny).*, vol. 476, pp. 491–504, 2019, doi: 10.1016/j.ins.2018.06.017.

[7] A. H. Farooqi, F. A. Khan, J. Wang, and S. Lee, "A novel intrusion detection framework for wireless sensor networks," *Pers. Ubiquitous Comput.*, vol. 17, no. 5, pp. 907–919, 2013, doi: 10.1007/s00779-012-0529-y.

[8] B. Blywis, "A real-time and energy-efficient MAC protocol for wireless sensor networks Pardeep Kumar *, Mesut Güne ş , Qasim Mushtaq and," vol. 1, no. 2, pp. 128–142, 2009.

[9] W. Choi, P. Shah, and S. K. Das, "A framework for energy-saving data gathering using two-phase clustering in wireless sensor networks," *Proc. MOBIQUITOUS 2004 - 1st*

Annu. Int. Conf. Mob. Ubiquitous Syst. Netw. Serv., pp. 203–212, 2004, doi: 10.1109/mobiq.2004.1331727.

[10] S. Messous and H. Liouane, “Online sequential DV-hop localization algorithm for wireless sensor networks,” *Mob. Inf. Syst.*, vol. 2020, 2020, doi: 10.1155/2020/8195309.

[11] G. Farjamnia, Y. Gasimov, and C. Kazimov, “An Improved DV-Hop for Detecting Wormhole Attacks in Wireless Sensor Networks,” vol. 9, no. 1, pp. 1–24, 2020.

[12] S. T. Patel, “$ 5Hylhz 6 \ Elo $ Wwdfn ’ Hwhfwlrq 7Hfkqltxhv Lq : 61,” vol. 17, pp. 4–8.

[13] J. Jiang, G. Han, C. Zhu, Y. Dong, and N. Zhang, “Secure localization in wireless sensor networks: A survey,” *J. Commun.*, vol. 6, no. 6, pp. 460–470, 2011, doi: 10.4304/jcm.6.6.460-470.

[14] B. Madagouda and R. Sumathi, “Artificial Neural Network Approach using Mobile Agent for Localization in Wireless Sensor Networks,” *Adv. Sci. Technol. Eng. Syst. J.*, vol. 6, no. 1, pp. 1137–1144, 2021, doi: 10.25046/aj0601127.

[15] “Wireless Sensor Network (WSN) Architecture And Applications.”

[16] T. H. Kim *et al.*, “A Novel Trust Evaluation Process for Secure Localization Using a Decentralized Blockchain in Wireless Sensor Networks,” *IEEE Access*, vol. 7, pp. 184133–184144, 2019, doi: 10.1109/ACCESS.2019.2960609.

[17] A. Sagu, N. S. Gill, and P. Gulia, “Hybrid Deep Neural Network Model for Detection of Security Attacks in IoT Enabled Environment,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 1, pp. 120–127, 2022, doi: 10.14569/IJACSA.2022.0130115.

[18] R. Bharathi, S. Kannadhasan, B. Padminidevi, M. S. Maharajan, R. Nagarajan, and M. M. Tonmoy, “Predictive Model Techniques with Energy Efficiency for IoT-Based Data Transmission in Wireless Sensor Networks,” vol. 2022, 2022.

[19] Z. Abubaker, N. Javaid, A. Almogren, M. Akbar, M. Zuair, and J. Ben-Othman, “Blockchained service provisioning and malicious node detection via federated learning in scalable Internet of Sensor Things networks,” *Comput. Networks*, vol. 204, no. January, p. 108691, 2022, doi: 10.1016/j.comnet.2021.108691.

[20] C. Ming, C. Xiaoting, D. Wensheng, and G. Jiahui, “A secure blockchain - based group key agreement protocol for IoT,” *J. Supercomput.*, vol. 77, no. 8, pp. 9046–9068, 2021, doi: 10.1007/s11227-020-03561-y.

[21] I. G. A. Poornima and B. Paramasivan, “Anomaly detection in wireless sensor network using machine learning algorithm,” *Comput. Commun.*, vol. 151, no. January, pp. 331–

337, 2020, doi: 10.1016/j.comcom.2020.01.005.

[22] M. S. Yousefpoor and H. Barati, “Dynamic key management algorithms in wireless sensor networks: A survey,” *Comput. Commun.*, vol. 134, no. November 2018, pp. 52–69, 2019, doi: 10.1016/j.comcom.2018.11.005.

[23] J. Grover and S. Sharma, “Security issues in Wireless Sensor Network-A review,” *2016 5th Int. Conf. Reliab. Infocom Technol. Optim. ICRITO 2016 Trends Futur. Dir.*, pp. 397–404, 2016, doi: 10.1109/ICRITO.2016.7784988.

[24] M. N. U. Islam, A. Fahmin, M. S. Hossain, and M. Atiquzzaman, “Denial-of-Service Attacks on Wireless Sensor Network and Defense Techniques,” *Wirel. Pers. Commun.*, vol. 116, no. 3, pp. 1993–2021, 2021, doi: 10.1007/s11277-020-07776-3.

[25] O. A. Osanaiye, A. S. Alfa, and G. P. Hancke, “Denial of Service Defence for Resource Availability in Wireless Sensor Networks,” *IEEE Access*, vol. 6, pp. 6975–7004, 2018, doi: 10.1109/ACCESS.2018.2793841.

[26] M. Noman Riaz, A. Buriro, and A. Mahboob, “Classification of Attacks on Wireless Sensor Networks: A Survey,” *Int. J. Wirel. Microw. Technol.*, vol. 8, no. 6, pp. 15–39, 2018, doi: 10.5815/ijwmt.2018.06.02.

[27] B. Ahmad, W. Jian, Z. A. Ali, S. Tanvir, and M. S. A. Khan, “Hybrid Anomaly Detection by Using Clustering for Wireless Sensor Network,” *Wirel. Pers. Commun.*, vol. 106, no. 4, pp. 1841–1853, 2019, doi: 10.1007/s11277-018-5721-6.

[28] M. Wazid and A. K. Das, “An Efficient Hybrid Anomaly Detection Scheme Using K-Means Clustering for Wireless Sensor Networks,” *Wirel. Pers. Commun.*, vol. 90, no. 4, pp. 1971–2000, 2016, doi: 10.1007/s11277-016-3433-3.

[29] R. Singh, J. Singh, and R. Singh, “Fuzzy based advanced hybrid intrusion detection system to detect malicious nodes in wireless sensor networks,” *Wirel. Commun. Mob. Comput.*, vol. 2017, 2017, doi: 10.1155/2017/3548607.

[30] P. P. Devi and B. Jaison, “Protection on Wireless Sensor Network from Clone Attack using the SDN-Enabled Hybrid Clone Node Detection Mechanisms,” *Comput. Commun.*, vol. 152, no. January, pp. 316–322, 2020, doi: 10.1016/j.comcom.2020.01.064.

[31] Y. Lyu, Y. Mo, S. Yue, and W. Liu, “Improved Beetle Antennae Algorithm Based on Localization for Jamming Attack in Wireless Sensor Networks,” *IEEE Access*, vol. 10, pp. 13071–13088, 2022, doi: 10.1109/ACCESS.2022.3146431.

[32] A. Keramatpour, A. Nikanjam, and H. Ghaffarian, “Deployment of Wireless Intrusion

Detection Systems to Provide the Most Possible Coverage in Wireless Sensor Networks Without Infrastructures," *Wirel. Pers. Commun.*, vol. 96, no. 3, pp. 3965–3978, 2017, doi: 10.1007/s11277-017-4363-4.

[33] Ö. Cepheli, S. Büyükçorak, and G. Karabulut Kurt, "Hybrid Intrusion Detection System for DDoS Attacks," *J. Electr. Comput. Eng.*, vol. 2016, 2016, doi: 10.1155/2016/1075648.

[34] L. Gandhimathi and G. Murugaboopathi, "A Novel Hybrid Intrusion Detection Using Flow-Based Anomaly Detection and Cross-Layer Features in Wireless Sensor Network," *Autom. Control Comput. Sci.*, vol. 54, no. 1, pp. 62–69, 2020, doi: 10.3103/S0146411620010046.

[35] L. Moulad, H. Belhadaoui, and M. Rifi, "Implementation of an Hierarchical Hybrid Intrusion Detection Mechanism in Wireless Sensor Network Based on Energy Management," *Adv. Intell. Syst. Comput.*, vol. 756, pp. 360–377, 2019, doi: 10.1007/978-3-319-91337-7_33.

[36] A. Abduvaliyev, S. Lee, and Y. K. Lee, "Energy efficient hybrid intrusion detection system for wireless sensor networks," *ICEIE 2010 - 2010 Int. Conf. Electron. Inf. Eng. Proc.*, vol. 2, no. Iceie, pp. 25–29, 2010, doi: 10.1109/ICEIE.2010.5559708.

[37] R. M. Swarna Priya *et al.*, "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture," *Comput. Commun.*, vol. 160, no. May, pp. 139–149, 2020, doi: 10.1016/j.comcom.2020.05.048.

[38] P. R. Kanna and P. Santhi, "Hybrid Intrusion Detection using MapReduce based Black Widow Optimized Convolutional Long Short-Term Memory Neural Networks," *Expert Syst. Appl.*, vol. 194, no. October 2021, p. 116545, 2022, doi: 10.1016/j.eswa.2022.116545.

[39] P. S. Moon and P. K. Ingole, "An overview on: Intrusion detection system with secure hybrid mechanism in wireless sensor network," *Conf. Proceeding - 2015 Int. Conf. Adv. Comput. Eng. Appl. ICACEA 2015*, pp. 272–277, 2015, doi: 10.1109/ICACEA.2015.7164714.

[40] R. Singh, J. Singh, and R. Singh, "WRHT: A Hybrid Technique for Detection of Wormhole Attack in Wireless Sensor Networks," *Mob. Inf. Syst.*, vol. 2016, 2016, doi: 10.1155/2016/8354930.

[41] L. Yang, A. Moubayed, and A. Shami, "MTH-IDS: A Multitiered Hybrid Intrusion Detection System for Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 616–

632, 2022, doi: 10.1109/JIOT.2021.3084796.

[42] Y. Maleh, A. Ezzati, Y. Qasmaoui, and M. Mbida, “A global hybrid intrusion detection system for Wireless Sensor Networks,” *Procedia Comput. Sci.*, vol. 52, no. 1, pp. 1047–1052, 2015, doi: 10.1016/j.procs.2015.05.108.

[43] P. P. Devi and B. Jaison, “Protection on Wireless Sensor Network from Clone Attack using the SDN-Enabled Hybrid Clone Node Detection Mechanisms,” *Comput. Commun.*, vol. 152, no. September 2019, pp. 316–322, 2020, doi: 10.1016/j.comcom.2020.01.064.

[44] S. Saif, P. Das, S. Biswas, M. Khari, and V. Shanmuganathan, “HIIDS: Hybrid intelligent intrusion detection system empowered with machine learning and metaheuristic algorithms for application in IoT based healthcare,” *Microprocess. Microsyst.*, p. 104622, 2022, doi: 10.1016/j.micpro.2022.104622.

[45] M. Rabbani, Y. L. Wang, R. Khoshkangini, H. Jelodar, R. Zhao, and P. Hu, “A hybrid machine learning approach for malicious behaviour detection and recognition in cloud computing,” *J. Netw. Comput. Appl.*, vol. 151, no. May 2019, p. 102507, 2020, doi: 10.1016/j.jnca.2019.102507.

[46] S. M. Kumar, “Hybrid Optimized Deep Neural Network with Enhanced Conditional Random Field Based Intrusion Detection on Wireless Sensor Network,” *Neural Process. Lett.*, 2022, doi: 10.1007/s11063-022-10892-9.

[47] S. Mahajan, R. Harikrishnan, and K. Kotecha, “Prediction of Network Traffic in Wireless Mesh Networks Using Hybrid Deep Learning Model,” *IEEE Access*, vol. 10, pp. 7003–7015, 2022, doi: 10.1109/ACCESS.2022.3140646.

[48] J. Al Faysal *et al.*, “XGB-RF: A Hybrid Machine Learning Approach for IoT Intrusion Detection,” *Telecom*, vol. 3, no. 1, pp. 52–69, 2022, doi: 10.3390/telecom3010003.

[49] M. I. Alghamdi, “A Hybrid Model for Intrusion Detection in IoT Applications,” *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022, doi: 10.1155/2022/4553502.

[50] F. Sadikin, T. van Deursen, and S. Kumar, “A ZigBee Intrusion Detection System for IoT using Secure and Efficient Data Collection,” *Internet of Things*, vol. 12, p. 100306, 2020, doi: 10.1016/j.iot.2020.100306.

[51] P. Sun *et al.*, “DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system,” *Secur. Commun. Networks*, vol. 2020, 2020, doi: 10.1155/2020/8890306.

[52] P. Biswas, R. Charitha, S. Gavel, and A. S. Raghuvanshi, “Fault detection using hybrid of

KF-ELM for wireless sensor networks," *Proc. Int. Conf. Trends Electron. Informatics, ICOEI 2019*, no. Icoei, pp. 746–750, 2019, doi: 10.1109/ICOEI.2019.8862687.

[53] T. Rose, K. Kifayat, S. Abbas, and M. Asim, "A hybrid anomaly-based intrusion detection system to improve time complexity in the Internet of Energy environment," *J. Parallel Distrib. Comput.*, vol. 145, pp. 124–139, 2020, doi: 10.1016/j.jpdc.2020.06.012.

[54] W. Liu, J. Cheng, X. Wang, X. Lu, and J. Yin, "Hybrid differential privacy based federated learning for Internet of Things," *J. Syst. Archit.*, vol. 124, no. July 2021, p. 102418, 2022, doi: 10.1016/j.sysarc.2022.102418.

[55] S. K. Gupta, M. Tripathi, and J. Grover, "Hybrid optimization and deep learning based intrusion detection system," *Comput. Electr. Eng.*, vol. 100, no. March, p. 107876, 2022, doi: 10.1016/j.compeleceng.2022.107876.

[56] B. Cao, C. Li, Y. Song, Y. Qin, and C. Chen, "applied sciences Network Intrusion Detection Model Based on CNN and GRU," 2022.

[57] S. Ullah *et al.*, "HDL-IDS: A Hybrid Deep Learning Architecture for Intrusion Detection in the Internet of Vehicles," *Sensors*, vol. 22, no. 4, pp. 1–20, 2022, doi: 10.3390/s22041340.

[58] M. Wazid, "Hybrid Anomaly Detection using K-Means Clustering in Wireless Sensor Networks," *Research.Iiit.Ac.in*, p. 4, 2014.

[59] C. Umarani and S. Kannan, "Intrusion detection system using hybrid tissue growing algorithm for wireless sensor network," *Peer-to-Peer Netw. Appl.*, vol. 13, no. 3, pp. 752–761, 2020, doi: 10.1007/s12083-019-00781-9.

[60] C. Yin, S. Zhang, Z. Yin, and J. Wang, "Anomaly detection model based on data stream clustering," *Cluster Comput.*, vol. 22, no. s1, pp. 1729–1738, 2019, doi: 10.1007/s10586-017-1066-2.

[61] L. Chelouah, F. Semchedine, and L. Bouallouche-Medjkoune, "Localization protocols for mobile wireless sensor networks: A survey," *Comput. Electr. Eng.*, vol. 71, pp. 733–751, 2018, doi: 10.1016/j.compeleceng.2017.03.024.

[62] A. Hadir, K. Zine-Dine, M. Bakhouya, and J. El Kafi, "An improved DV-Hop localization algorithm for wireless sensor networks," *Int. Conf. Next Gener. Networks Serv. NGNS*, pp. 330–334, 2014, doi: 10.1109/NGNS.2014.6990273.

[63] I. Almomani, B. Al-Kasasbeh, and M. Al-Akhras, "WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks," *J. Sensors*, vol. 2016, 2016, doi:

10.1155/2016/4731953.

[64] O. Cheikhrouhou and A. Koubaa, “BlockLoc: Secure localization in the internet of things using blockchain,” *2019 15th Int. Wirel. Commun. Mob. Comput. Conf. IWCMC 2019*, pp. 629–634, 2019, doi: 10.1109/IWCMC.2019.8766440.

[65] S. T. Patel and N. H. Mistry, “A review: Sybil attack detection techniques in WSN,” *Proc. 2017 4th Int. Conf. Electron. Commun. Syst. ICECS 2017*, vol. 17, pp. 184–188, 2017, doi: 10.1109/ECS.2017.8067865.

[66] F. Y. Yavuz, D. Ünal, and E. Gül, “Deep learning for detection of routing attacks in the internet of things,” *Int. J. Comput. Intell. Syst.*, vol. 12, no. 1, pp. 39–58, 2018, doi: 10.2991/ijcis.2018.25905181.

[67] V. Sujatha and E. A. M. Anita, “FEM-hybrid machine learning approach for the detection of sybil attacks in the wireless sensor networks,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 7, pp. 1171–1179, 2019.

[68] X. Qi, X. Liu, and L. Liu, “A Combined Localization Algorithm for Wireless Sensor Networks,” *Math. Probl. Eng.*, vol. 2018, 2018, doi: 10.1155/2018/4648109.

[69] P. Li, X. Yu, H. Xu, J. Qian, L. Dong, and H. Nie, “Research on secure localization model based on trust valuation in wireless sensor networks,” *Secur. Commun. Networks*, vol. 2017, 2017, doi: 10.1155/2017/6102780.

[70] L. Song, L. Zhao, and J. Ye, “DV-Hop Node Location Algorithm Based on GSO in Wireless Sensor Networks,” *J. Sensors*, vol. 2019, 2019, doi: 10.1155/2019/2986954.

[71] M. Saud Khan and N. M. Khan, “Low Complexity Signed Response Based Sybil Attack Detection Mechanism in Wireless Sensor Networks,” *J. Sensors*, vol. 2016, 2016, doi: 10.1155/2016/9783072.

[72] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, S. Garg, and M. M. Hassan, “A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network,” *J. Parallel Distrib. Comput.*, vol. 164, pp. 55–68, 2022, doi: 10.1016/j.jpdc.2022.01.030.

[73] R. Goyat, G. Kumar, M. K. Rai, R. Saha, R. Thomas, and T. H. Kim, “Blockchain Powered Secure Range-Free Localization in Wireless Sensor Networks,” *Arab. J. Sci. Eng.*, vol. 45, no. 8, pp. 6139–6155, 2020, doi: 10.1007/s13369-020-04493-8.

[74] S. Awan, N. Javaid, S. Ullah, A. U. Khan, A. M. Qamar, and J. G. Choi, “Blockchain Based Secure Routing and Trust Management in Wireless Sensor Networks†,” *Sensors*, vol. 22, no. 2, pp. 1–24, 2022, doi: 10.3390/s22020411.

[75] H. H. Pajooh, M. Rashid, F. Alam, and S. Demidenko, “Multi-layer blockchain-based security architecture for internet of things,” *Sensors (Switzerland)*, vol. 21, no. 3, pp. 1–26, 2021, doi: 10.3390/s21030772.

[76] S. Otoum, I. Al Ridhawi, and H. Mouftah, “Securing Critical IoT Infrastructures with Blockchain-Supported Federated Learning,” *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2592–2601, 2022, doi: 10.1109/JIOT.2021.3088056.

[77] W. She, Q. Liu, Z. Tian, J. Sen Chen, B. Wang, and W. Liu, “Blockchain trust model for malicious node detection in wireless sensor networks,” *IEEE Access*, vol. 7, pp. 38947–38956, 2019, doi: 10.1109/ACCESS.2019.2902811.

[78] K. Fan *et al.*, “Blockchain-based secure time protection scheme in IoT,” *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4671–4679, 2019, doi: 10.1109/JIOT.2018.2874222.

[79] O. Friha, M. A. Ferrag, L. Shu, L. Maglaras, K. K. R. Choo, and M. Nafaa, “FELIDS: Federated learning-based intrusion detection system for agricultural Internet of Things,” *J. Parallel Distrib. Comput.*, vol. 165, pp. 17–31, 2022, doi: 10.1016/j.jpdc.2022.03.003.

[80] N. Javaid, “A Secure and Efficient Trust Model for Wireless Sensor IoTs Using Blockchain,” *IEEE Access*, vol. 10, pp. 4568–4579, 2022, doi: 10.1109/ACCESS.2022.3140401.

[81] D. Wu and N. Ansari, “A trust-evaluation-enhanced blockchain-secured industrial iot system,” *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5510–5517, 2021, doi: 10.1109/JIOT.2020.3030689.

[82] A. U. Khan, M. B. E. Sajid, A. Rauf, M. N. Saqib, F. Zaman, and N. Javaid, “Exploiting Blockchain and RMCV-Based Malicious Node Detection in ETD-LEACH for Wireless Sensor Networks,” *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022, doi: 10.1155/2022/7281872.

[83] R. Goyat, G. Kumar, M. Alazab, R. Saha, R. Thomas, and M. K. Rai, “A secure localization scheme based on trust assessment for WSNs using blockchain technology,” *Futur. Gener. Comput. Syst.*, vol. 125, pp. 221–231, 2021, doi: 10.1016/j.future.2021.06.039.

[84] D. Minoli and B. Occhiogrosso, “Blockchain mechanisms for IoT security,” *Internet of Things (Netherlands)*, vol. 1–2, pp. 1–13, 2018, doi: 10.1016/j.iot.2018.05.002.

[85] Z. Ma, L. Wang, and W. Zhao, “Blockchain-Driven Trusted Data Sharing with Privacy Protection in IoT Sensor Network,” *IEEE Sens. J.*, vol. 21, no. 22, pp. 25472–25479, 2021,

doi: 10.1109/JSEN.2020.3046752.

[86] X. Yang, Y. Chen, X. Qian, T. Li, and X. Lv, “BCEAD: A Blockchain-Empowered Ensemble Anomaly Detection for Wireless Sensor Network via Isolation Forest,” *Secur. Commun. Networks*, vol. 2021, 2021, doi: 10.1155/2021/9430132.

[87] M. Sarhan, W. W. Lo, S. Layeghy, and M. Portmann, “HBFL: A Hierarchical Blockchain-based Federated Learning Framework for a Collaborative IoT Intrusion Detection,” no. Ml, pp. 1–18, 2022, [Online]. Available: http://arxiv.org/abs/2204.04254

[88] S. J. Hsiao and W. T. Sung, “Employing Blockchain Technology to Strengthen Security of Wireless Sensor Networks,” *IEEE Access*, vol. 9, pp. 72326–72341, 2021, doi: 10.1109/ACCESS.2021.3079708.

[89] S. Abbas, N. Javaid, A. Almogren, S. M. Gulfam, A. Ahmed, and A. Radwan, “Securing Genetic Algorithm Enabled SDN Routing for Blockchain Based Internet of Things,” *IEEE Access*, vol. 9, pp. 139739–139754, 2021, doi: 10.1109/ACCESS.2021.3118948.

[90] E. H. Abualsauod, “A hybrid blockchain method in internet of things for privacy and security in unmanned aerial vehicles network,” *Comput. Electr. Eng.*, vol. 99, no. October 2021, p. 107847, 2022, doi: 10.1016/j.compeleceng.2022.107847.

[91] B. Bhushan and G. Sahoo, “Recent advances in attacks, technical challenges, vulnerabilities and their countermeasures in wireless sensor networks,” *Wirel. Pers. Commun.*, vol. 98, no. 2, pp. 2037–2077, 2018, doi: 10.1007/s11277-017-4962-0.

[92] S. Gomathi and C. Gopala Krishnan, “Malicious Node Detection in Wireless Sensor Networks Using an Efficient Secure Data Aggregation Protocol,” *Wirel. Pers. Commun.*, vol. 113, no. 4, pp. 1775–1790, 2020, doi: 10.1007/s11277-020-07291-5.

[93] M. Kaur and A. Munjal, “Data aggregation algorithms for wireless sensor network: A review,” *Ad Hoc Networks*, vol. 100, 2020, doi: 10.1016/j.adhoc.2020.102083.

[94] A. A. Jasim *et al.*, “Secure and Energy-Efficient Data Aggregation Method Based on an Access Control Model,” *IEEE Access*, vol. 7, pp. 164327–164343, 2019, doi: 10.1109/ACCESS.2019.2952904.

[95] X. Qi, X. Liu, J. Yu, and Q. Zhang, “A Privacy Data Aggregation Scheme for Wireless Sensor Networks,” *Procedia Comput. Sci.*, vol. 174, no. 2019, pp. 578–583, 2020, doi: 10.1016/j.procs.2020.06.127.

[96] K. P. Uvarajan and C. Gowri Shankar, “An Integrated Trust Assisted Energy Efficient Greedy Data Aggregation for Wireless Sensor Networks,” *Wirel. Pers. Commun.*, vol. 114,

no. 1, pp. 813–833, 2020, doi: 10.1007/s11277-020-07394-z.

[97] A. Razaque and S. S. Rizvi, "Secure data aggregation using access control and authentication for wireless sensor networks," *Comput. Secur.*, vol. 70, no. July, pp. 532–545, 2017, doi: 10.1016/j.cose.2017.07.001.

[98] X. Liu, J. Yu, F. Li, W. Lv, Y. Wang, and X. Cheng, "Data Aggregation in Wireless Sensor Networks: From the Perspective of Security," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6495–6513, 2020, doi: 10.1109/JIOT.2019.2957396.

[99] Z. Zhang, J. Li, and X. Yang, "Data Aggregation in Heterogeneous Wireless Sensor Networks by Using Local Tree Reconstruction Algorithm," *Complexity*, vol. 2020, 2020, doi: 10.1155/2020/3594263.

[100] C. Bekara, M. Laurent-Maknavicius, and K. Bekara, "SAPC: A secure aggregation protocol for cluster-based wireless sensor networks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4864 LNCS, pp. 784–798, 2007, doi: 10.1007/978-3-540-77024-4_71.

[101] R. Maivizhi and P. Yogesh, "Q-learning based routing for in-network aggregation in wireless sensor networks," *Wirel. Networks*, vol. 8, 2021, doi: 10.1007/s11276-021-02564-8.

[102] M. Mathapati, T. S. Kumaran, K. H. S. Prasad, and K. Patil, "Framework with temporal attribute for secure data aggregation in sensor network," *SN Appl. Sci.*, vol. 2, no. 12, 2020, doi: 10.1007/s42452-020-03773-0.

[103] M. Naghibi and H. Barati, "SHSDA: secure hybrid structure data aggregation method in wireless sensor networks," *J. Ambient Intell. Humaniz. Comput.*, no. 0123456789, 2021, doi: 10.1007/s12652-020-02751-z.

[104] S. Gopikrishnan and P. Priakanth, "HSDA: hybrid communication for secure data aggregation in wireless sensor network," *Wirel. Networks*, vol. 22, no. 3, pp. 1061–1078, 2016, doi: 10.1007/s11276-015-1122-x.

[105] N. M. Saravana Kumar, E. Suryaprabha, and K. Hariprasath, "Machine learning based hybrid model for energy efficient secured transmission in wireless sensor networks," *J. Ambient Intell. Humaniz. Comput.*, 2021, doi: 10.1007/s12652-021-02946-y.

[106] D. B.D. and F. Al-Turjman, "A hybrid secure routing and monitoring mechanism in IoT-based wireless sensor networks," *Ad Hoc Networks*, vol. 97, 2020, doi: 10.1016/j.adhoc.2019.102022.

[107] N. Rouissi and H. Gharsellaoui, “Improved Hybrid LEACH Based Approach for Preserving Secured Integrity in Wireless Sensor Networks,” *Procedia Comput. Sci.*, vol. 112, pp. 1429–1438, 2017, doi: 10.1016/j.procs.2017.08.103.

[108] E. A. M. Anita, R. Geetha, and E. Kannan, “A Novel Hybrid Key Management Scheme for Establishing Secure Communication in Wireless Sensor Networks,” *Wirel. Pers. Commun.*, vol. 82, no. 3, pp. 1419–1433, 2015, doi: 10.1007/s11277-015-2290-9.

[109] P. Gao, “Mechatronics and Automatic Control Systems,” *Lect. Notes Electr. Eng.*, vol. 237, pp. 1063–1071, 2014, doi: 10.1007/978-3-319-01273-5.

[110] G. Gebreyesus, “Secure Intrusion Detection System for Hierarchically Distributed Wireless Sensor Networks,” pp. 9–14, 2021.

[111] R. S. Mangrulkar and P. D. Negandhi, “Applications of Machine Learning in Wireless Sensor Networks,” *Soft Comput. Wirel. Sens. Networks*, pp. 51–74, 2018, doi: 10.1201/9780429438639-3.

[112] N. Rouissi, H. Gharsellaoui, and S. Bouamama, “Improvement of watermarking-LEACH algorithm based on trust for wireless sensor networks,” *Procedia Comput. Sci.*, vol. 159, pp. 803–813, 2019, doi: 10.1016/j.procs.2019.09.239.

[113] A. Vinitha, M. S. S. Rukmini, and Dhirajsunehra, “Secure and energy aware multi-hop routing protocol in WSN using Taylor-based hybrid optimization algorithm,” *J. King Saud Univ. - Comput. Inf. Sci.*, no. xxxx, 2019, doi: 10.1016/j.jksuci.2019.11.009.

[114] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, “Explainable Artificial Intelligence (XAI) to Enhance Trust Management in Intrusion Detection Systems Using Decision Tree Model,” *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/6634811.

[115] W. Zhang, D. Han, K. C. Li, and F. I. Massetto, “Wireless sensor network intrusion detection system based on MK-ELM,” *Soft Comput.*, vol. 24, no. 16, pp. 12361–12374, 2020, doi: 10.1007/s00500-020-04678-1.

[116] S. Godala and R. P. V. Vaddella, “A study on intrusion detection system in wireless sensor networks,” *Int. J. Commun. Networks Inf. Secur.*, vol. 12, no. 1, pp. 127–141, 2020.

[117] S. A. Elsaid and N. S. Albatati, “An optimized collaborative intrusion detection system for wireless sensor networks,” *Soft Comput.*, vol. 24, no. 16, pp. 12553–12567, 2020, doi: 10.1007/s00500-020-04695-0.

[118] Y. Wang *et al.*, “An Exhaustive Research on the Application of Intrusion Detection Technology in Computer Network Security in Sensor Networks,” *J. Sensors*, vol. 2021,

2021, doi: 10.1155/2021/5558860.

[119] D. Praveen Kumar, T. Amgoth, and C. S. R. Annavarapu, “Machine learning algorithms for wireless sensor networks: A survey,” *Inf. Fusion*, vol. 49, pp. 1–25, Sep. 2019, doi: 10.1016/J.INFFUS.2018.09.013.

[120] Z. Al Aghbari, A. M. Khedr, W. Osamy, I. Arif, and D. P. Agrawal, “Routing in Wireless Sensor Networks Using Optimization Techniques: A Survey,” *Wirel. Pers. Commun.*, vol. 111, no. 4, pp. 2407–2434, 2020, doi: 10.1007/s11277-019-06993-9.

[121] D. Praveen Kumar, T. Amgoth, and C. S. R. Annavarapu, “Machine learning algorithms for wireless sensor networks: A survey,” *Inf. Fusion*, vol. 49, no. April 2018, pp. 1–25, 2019, doi: 10.1016/j.inffus.2018.09.013.

[122] P. J. B. Pajila, E. G. Julie, and Y. H. Robinson, *FBDR-Fuzzy Based DDoS Attack Detection and Recovery Mechanism for Wireless Sensor Networks*, no. 0123456789. Springer US, 2021. doi: 10.1007/s11277-021-09040-8.

[123] P. Ahlawat and M. Dave, “An attack model based highly secure key management scheme for wireless sensor networks,” *Procedia Comput. Sci.*, vol. 125, pp. 201–207, 2018, doi: 10.1016/j.procs.2017.12.028.

[124] A. Ismail and R. Amin, “Malicious Cluster Head Detection Mechanism in Wireless Sensor Networks,” *Wirel. Pers. Commun.*, vol. 108, no. 4, pp. 2117–2135, 2019, doi: 10.1007/s11277-019-06512-w.

[125] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, “A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends,” *Proc. IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016, doi: 10.1109/JPROC.2016.2558521.

[126] Y. Wu, D. Wei, and J. Feng, “Network attacks detection methods based on deep learning techniques: A survey,” *Secur. Commun. Networks*, vol. 2020, 2020, doi: 10.1155/2020/8872923.

[127] “1999 DARPA Intrusion Detection Evaluation Dataset | MIT Lincoln Laboratory.” https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset (accessed Jan. 08, 2021).

[128] “KDD Cup 1999 Data.” http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed Jan. 08, 2021).

[129] G. Meena and R. R. Choudhary, “A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA,” *2017 Int. Conf. Comput. Commun. Electron. COMPTELIX*

2017, pp. 553–558, 2017, doi: 10.1109/COMPTELIX.2017.8004032.

[130] S. Pande, A. Khamparia, and D. Gupta, "Feature selection and comparison of classification algorithms for wireless sensor networks," *J. Ambient Intell. Humaniz. Comput.*, no. 0123456789, 2021, doi: 10.1007/s12652-021-03411-6.

[131] S. A. V. Jatti and V. J. K. Kishor Sontif, "Intrusion detection systems," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2 Special Issue 11, pp. 3976–3983, 2019, doi: 10.35940/ijrte.B1540.0982S1119.

[132] "IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB." https://www.unb.ca/cic/datasets/ids-2017.html (accessed Aug. 30, 2021).

[133] A. Bansal and S. Kaur, *Extreme gradient boosting based tuning for classification in intrusion detection systems*, vol. 905. Springer Singapore, 2018. doi: 10.1007/978-981-13-1810-8_37.

[134] M. Anbarasan *et al.*, "Detection of flood disaster system based on IoT, big data and convolutional deep neural network," *Comput. Commun.*, vol. 150, no. November 2019, pp. 150–157, 2020, doi: 10.1016/j.comcom.2019.11.022.

[135] S. K. Gupta, M. Tripathi, and J. Grover, "Hybrid optimization and deep learning based intrusion detection system," *Comput. Electr. Eng.*, vol. 100, no. June 2021, p. 107876, 2022, doi: 10.1016/j.compeleceng.2022.107876.

[136] P. Roy and C. Chowdhury, "A Survey of Machine Learning Techniques for Indoor Localization and Navigation Systems," *J. Intell. Robot. Syst. Theory Appl.*, vol. 101, no. 3, 2021, doi: 10.1007/s10846-021-01327-z.

[137] S. Subramani, "Deep Learning based IDS for Secured Routing in Wireless Sensor Networks using Fuzzy Genetic Approach," 2022.

[138] S. Kim, H. Cai, C. Hua, P. Gu, W. Xu, and J. Park, "Collaborative Anomaly Detection for Internet of Things based on Federated Learning," *2020 IEEE/CIC Int. Conf. Commun. China, ICCC 2020*, no. Iccc, pp. 623–628, 2020, doi: 10.1109/ICCC49849.2020.9238913.

[139] Y. Kayode Saheed, A. Idris Abiodun, S. Misra, M. Kristiansen Holone, and R. Colomo-Palacios, "A machine learning-based intrusion detection for detecting internet of things network attacks," *Alexandria Eng. J.*, vol. 61, no. 12, pp. 9395–9409, 2022, doi: 10.1016/j.aej.2022.02.063.

[140] A. Intelligence, S. Science, B. Media, and S. Nature, "A new hybrid approach for intrusion detection using machine learning," pp. 2735–2761, 2019.

[141] Q. A. Al-Haija, M. Krichen, and W. A. Elhaija, “Machine-Learning-Based Darknet Traffic Detection System for IoT Applications,” *Electron.*, vol. 11, no. 4, 2022, doi: 10.3390/electronics11040556.

[142] D. Praveen Kumar, T. Amgoth, and C. S. R. Annavarapu, “Machine learning algorithms for wireless sensor networks: A survey,” *Inf. Fusion*, vol. 49, no. September 2018, pp. 1–25, 2019, doi: 10.1016/j.inffus.2018.09.013.

[143] G. Kocher and G. Kumar, “Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges,” *Soft Comput.*, vol. 25, no. 15, pp. 9731–9763, 2021, doi: 10.1007/s00500-021-05893-0.

[144] N. Chikhalia, D. Sahoo, and H. Mandali, “Hybrid Machine Learning Technique for Network Anomaly Detection”.

[145] D. Upadhyay, J. Manero, M. Zaman, and S. Sampalli, “Learning Classifiers for Intrusion Detection on Power Grids,” *Ieee Trans. Netw. Serv. Manag.*, vol. 18, no. 1, pp. 1104–1116, 2021.

[146] S. Taleb, A. Al Sallab, H. Hajj, Z. Dawy, R. Khanna, and A. Keshavamurthy, “Deep learning with ensemble classification method for sensor sampling decisions,” *2016 Int. Wirel. Commun. Mob. Comput. Conf. IWCMC 2016*, pp. 114–119, 2016, doi: 10.1109/IWCMC.2016.7577043.

[147] M. Sri Vidya and G. R. Sakthidharan, “Accurate Anomaly Detection using various Machine Learning methods for IoT devices in Indoor Environment,” *Proc. 5th Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud), I-SMAC 2021*, pp. 308–316, 2021, doi: 10.1109/I-SMAC52330.2021.9640962.

[148] M. Feurer and F. Hutter, “Hyperparameter Optimization,” pp. 3–33, 2019, doi: 10.1007/978-3-030-05318-5_1.

[149] N. Mohd, A. Singh, and H. S. Bhadauria, “A Novel SVM Based IDS for Distributed Denial of Sleep Strike in Wireless Sensor Networks,” *Wirel. Pers. Commun.*, vol. 111, no. 3, pp. 1999–2022, 2020, doi: 10.1007/s11277-019-06969-9.

[150] R. Panigrahi *et al.*, “Intrusion detection in cyber–physical environment using hybrid Naïve Bayes—Decision table and multi-objective evolutionary feature selection,” *Comput. Commun.*, vol. 188, no. September 2021, pp. 133–144, 2022, doi: 10.1016/j.comcom.2022.03.009.

[151] S. M. Kasongo, “An advanced intrusion detection system for IIoT Based on GA and tree

based algorithms," *IEEE Access*, vol. 9, pp. 113199–113212, 2021, doi: 10.1109/ACCESS.2021.3104113.

[152] M. F. Suleiman and B. Issac, "Performance Comparison of Intrusion Detection Machine Learning Classifiers on Benchmark and New Datasets," *28th Int. Conf. Comput. Theory Appl. ICCTA 2018 - Proc.*, pp. 19–23, 2018, doi: 10.1109/ICCTA45985.2018.9499140.

[153] G. H. Lai, "Detection of wormhole attacks on IPv6 mobility-based wireless sensor network," *Eurasip J. Wirel. Commun. Netw.*, vol. 2016, no. 1, 2016, doi: 10.1186/s13638-016-0776-0.

[154] Y. Yuan, L. Huo, Z. Wang, and D. Hogrefe, "Secure APIT Localization Scheme Against Sybil Attacks in Distributed Wireless Sensor Networks," *IEEE Access*, vol. 6, pp. 27629–27636, 2018, doi: 10.1109/ACCESS.2018.2836898.

[155] P. Nancy, S. Muthurajkumar, S. Ganapathy, S. V. N. Santhosh Kumar, M. Selvi, and K. Arputharaj, "Intrusion detection using dynamic feature selection and fuzzy temporal decision tree classification for wireless sensor networks," *IET Commun.*, vol. 14, no. 5, pp. 888–895, 2020, doi: 10.1049/iet-com.2019.0172.

[156] G. Qi, J. Zhou, W. Jia, M. Liu, S. Zhang, and M. Xu, "Intrusion Detection for Network Based on Elite Clone Artificial Bee Colony and Back Propagation Neural Network," *Wirel. Commun. Mob. Comput.*, vol. 2021, 2021, doi: 10.1155/2021/9956371.

[157] S. Jiang, J. Zhao, and X. Xu, "SLGBM: An intrusion detection mechanism for wireless sensor networks in smart environments," *IEEE Access*, vol. 8, pp. 169548–169558, 2020, doi: 10.1109/ACCESS.2020.3024219.

[158] M. Abdan and S. A. H. Seno, "Machine Learning Methods for Intrusive Detection of Wormhole Attack in Mobile Ad Hoc Network (MANET)," *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022, doi: 10.1155/2022/2375702.

[159] G. Farjamnia, Y. Gasimov, and C. Kazimov, "Review of the Techniques Against the Wormhole Attacks on Wireless Sensor Networks," *Wirel. Pers. Commun.*, vol. 105, no. 4, pp. 1561–1584, 2019, doi: 10.1007/s11277-019-06160-0.

[160] M. Mahajan, K. T. V. Reddy, and M. Rajput, "Design and Simulation of a Blacklisting Technique for Detection of Hello flood Attack on LEACH Protocol," *Procedia Comput. Sci.*, vol. 79, pp. 675–682, 2016, doi: 10.1016/j.procs.2016.03.086.

[161] U. Jain and M. Hussain, "Securing Wireless Sensors in Military Applications through Resilient Authentication Mechanism," *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 719–

728, 2020, doi: 10.1016/j.procs.2020.04.078.

[162] S. Z. Wang, Y. Li, and W. Cheng, “Distributed classification of localization attacks in sensor networks using exchange-based feature extraction and classifier,” *J. Sensors*, vol. 2016, 2016, doi: 10.1155/2016/8672305.

[163] A. Payal, C. S. Rai, and B. V. R. Reddy, “Artificial Neural Networks for developing localization framework in Wireless Sensor Networks,” *2014 Int. Conf. Data Min. Intell. Comput. ICDMIC 2014*, pp. 0–5, 2014, doi: 10.1109/ICDMIC.2014.6954228.

[164] R. Dela Cruz, “Artificial Neural Network-based Localization in Wireless Sensor Networks Artificial Neural Network-based Localization in Wireless Sensor Networks,” no. November 2018, pp. 0–14, 2019.

[165] J. Jiang, G. Han, H. Wang, and M. Guizani, “A survey on location privacy protection in Wireless Sensor Networks,” *J. Netw. Comput. Appl.*, vol. 125, no. October 2018, pp. 93–114, 2019, doi: 10.1016/j.jnca.2018.10.008.

[166] R. Singh, J. Singh, and R. Singh, “A Novel Sybil Attack Detection Technique for Wireless Sensor Networks,” *Adv. Comput. Sci. Technol.*, vol. 10, no. 2, pp. 185–202, 2017.

[167] Z. Han *et al.*, “CNN-Based Attack Defense for Device-Free Localization,” *Mob. Inf. Syst.*, vol. 2022, 2022, doi: 10.1155/2022/2323293.

[168] J. Chen, S. H. Sackey, J. H. Anajemba, X. Zhang, and Y. He, “Energy-Efficient Clustering and Localization Technique Using Genetic Algorithm in Wireless Sensor Networks,” *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/5541449.

[169] H. Chen, W. Lou, Z. Wang, J. Wu, Z. Wang, and A. Xi, “Securing DV-Hop localization against wormhole attacks in wireless sensor networks,” *Pervasive Mob. Comput.*, vol. 16, no. PA, pp. 22–35, 2015, doi: 10.1016/j.pmcj.2014.01.007.

[170] G. G. Gebremariam, J. Panada, S. Indu, and M. B. Road, “Localization and Detection of Multiple Attacks in Wireless Sensor Networks Using 1 Introduction”.

[171] F. Khelifi, A. Bradai, A. Benslimane, P. Rawat, and M. Atri, “A Survey of Localization Systems in Internet of Things,” *Mob. Networks Appl.*, vol. 24, no. 3, pp. 761–785, 2019, doi: 10.1007/s11036-018-1090-3.

[172] O. Cheikhrouhou, G. M. Bhatti, and R. Alroobaea, “A hybrid DV-hop algorithm using RSSI for localization in large-scale wireless sensor networks,” *Sensors (Switzerland)*, vol. 18, no. 5, pp. 1–14, 2018, doi: 10.3390/s18051469.

[173] D. Praveen Kumar, T. Amgoth, and C. S. R. Annavarapu, “Machine learning algorithms

for wireless sensor networks: A survey," *Inf. Fusion*, vol. 49, pp. 1–25, 2019, doi: 10.1016/j.inffus.2018.09.013.

[174] R. Ahmad, R. Wazirali, and T. Abu-Ain, "Machine Learning for Wireless Sensor Networks Security: An Overview of Challenges and Issues," *Sensors*, vol. 22, no. 13. 2022. doi: 10.3390/s22134730.

[175] D. Zhou, W. Liu, W. Zhou, and S. Dong, "Research on network traffic identification based on multi layer perceptron," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 12, no. 1, pp. 201–208, 2014, doi: 10.12928/TELKOMNIKA.v12i1.1051.

[176] M. A. Alsheikh, S. Lin, D. Niyato, and H. P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014, doi: 10.1109/COMST.2014.2320099.

[177] W. He, F. Lu, J. Chen, Y. Ruan, T. Lu, and Y. Zhang, "A Kernel-Based Node Localization in Anisotropic Wireless Sensor Network," *Sci. Program.*, vol. 2021, no. 1, 2021, doi: 10.1155/2021/9944358.

[178] B. K. Madagouda and R. Sumathi, "Analysis of Localization Using ANN Models in Wireless Sensor Networks," *2019 IEEE Pune Sect. Int. Conf. PuneCon 2019*, pp. 18–21, 2019, doi: 10.1109/PuneCon46936.2019.9105871.

[179] N. Moustafa, G. Creech, and J. Slay, "Anomaly detection system using beta mixture models and outlier detection," in *Advances in Intelligent Systems and Computing*, 2018, vol. 710, pp. 125–135. doi: 10.1007/978-981-10-7871-2_13.

[180] "cybersecurity_attacks | Kaggle." https://www.kaggle.com/iamranjann/cybersecurity-attacks (accessed Mar. 11, 2021).

[181] N. Moustafa, G. Creech, and J. Slay, "Flow aggregator module for analysing network traffic," in *Advances in Intelligent Systems and Computing*, 2018, vol. 710, pp. 19–29. doi: 10.1007/978-981-10-7871-2_3.

[182] "The UNSW-NB15 data set description." https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/ (accessed Mar. 11, 2021).

[183] N. Moustafa and J. Slay, "A network forensic scheme using correntropy-variation for attack detection," in *IFIP Advances in Information and Communication Technology*, 2018, vol. 532, pp. 225–239. doi: 10.1007/978-3-319-99277-8_13.

[184] N. Moustafa, G. Creech, and J. Slay, "Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models," 2017, pp. 127–156.

doi: 10.1007/978-3-319-59439-2_5.

[185] N. Moustafa and J. Slay, “The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set,” *Inf. Secur. J.*, vol. 25, no. 1–3, pp. 18–31, Apr. 2016, doi: 10.1080/19393555.2015.1125974.

[186] C. R. Panigrahi, *Advanced Computing and Intelligent Engineering*, vol. 1. 2018.

[187] M. N. A. Shaon and K. Ferens, “A computationally intelligent approach to the detection of wormhole attacks in wireless sensor networks,” *Adv. Sci. Technol. Eng. Syst.*, vol. 2, no. 3, pp. 302–320, 2017, doi: 10.25046/aj020340.

[188] M. Mittal, R. P. de Prado, Y. Kawai, S. Nakajima, and J. E. Muñoz-Expósito, “Machine learning techniques for energy efficiency and anomaly detection in hybrid wireless sensor networks,” *Energies*, vol. 14, no. 11, pp. 1–21, 2021, doi: 10.3390/en14113125.

[189] V. Hassija *et al.*, “A blockchain and deep neural networks-based secure framework for enhanced crop protection,” *Ad Hoc Networks*, vol. 119, no. April, p. 102537, 2021, doi: 10.1016/j.adhoc.2021.102537.

[190] K. Ncibi, T. Sadraoui, M. Faycel, and A. Djenina, “A Multilayer Perceptron Artificial Neural Networks Based a Preprocessing and Hybrid Optimization Task for Data Mining and Classification,” *Int. J. Econom. Financ. Manag. Vol. 5, 2017, Pages 12-21*, vol. 5, no. 1, pp. 12–21, 2017, doi: 10.12691/ijefm-5-1-3.

[191] M. Al-Imran and S. H. Ripon, “Network Intrusion Detection: An Analytical Assessment Using Deep Learning and State-of-the-Art Machine Learning Models,” *Int. J. Comput. Intell. Syst.*, vol. 14, no. 1, pp. 1–20, 2021, doi: 10.1007/s44196-021-00047-4.

[192] D. Kothona, I. P. Panapakidis, and G. C. Christoforidis, “A novel hybrid ensemble LSTM-FFNN forecasting model for very short-term and short-term PV generation forecasting,” *IET Renew. Power Gener.*, vol. 16, no. 1, pp. 3–18, 2022, doi: 10.1049/rpg2.12209.

[193] K. P. S. Kumar, S. A. H. Nair, D. Guha Roy, B. Rajalingam, and R. S. Kumar, “Security and privacy-aware Artificial Intrusion Detection System using Federated Machine Learning,” *Comput. Electr. Eng.*, vol. 96, no. PA, p. 107440, 2021, doi: 10.1016/j.compeleceng.2021.107440.

[194] Z. Tan, A. Jamdagni, P. Nanda, X. He, and R. P. Liu, “Evaluation on multivariate correlation analysis based denial-of-service attack detection system,” *ACM Int. Conf. Proceeding Ser.*, pp. 160–164, 2012, doi: 10.1145/2490428.2490450.

[195] Z. Tan, “Detection of Denial-of-Service Attacks Based on By,” no. December, pp. 1–14, 2013.

[196] M. Nivaashini and P. Thangaraj, “Computational intelligence techniques for automatic detection of Wi-Fi attacks in wireless IoT networks,” *Wirel. Networks*, vol. 27, no. 4, pp. 2761–2784, 2021, doi: 10.1007/s11276-021-02594-2.

[197] M. Farooq-I-Azam, Q. Ni, and E. A. Ansari, “Intelligent Energy Efficient Localization Using Variable Range Beacons in Industrial Wireless Sensor Networks,” *IEEE Trans. Ind. Informatics*, vol. 12, no. 6, pp. 2206–2216, 2016, doi: 10.1109/TII.2016.2606084.

[198] S. Karagol and D. Yildiz, “A Novel Path Planning Model Based on Nested Regular Hexagons for Mobile Anchor-Assisted Localization in Wireless Sensor Networks,” *Arab. J. Sci. Eng.*, 2022, doi: 10.1007/s13369-021-06374-0.

[199] Y. Alsultanny, “Machine Learning by Data Mining REPTree and M5P for Predicating Novel Information for PM10,” *Cloud Comput. Data Sci.*, pp. 40–48, 2020, doi: 10.37256/ccds.112020418.

[200] A. Hadir, Y. Regragui, and N. M. Garcia, “Accurate Range-Free Localization Algorithms Based on PSO for Wireless Sensor Networks,” *IEEE Access*, vol. 9, pp. 149906–149924, 2021, doi: 10.1109/ACCESS.2021.3123360.

[201] G. Kumar, M. K. Rai, H. J. Kim, and R. Saha, “A Secure Localization Approach Using Mutual Authentication and Insider Node Validation in Wireless Sensor Networks,” *Mob. Inf. Syst.*, vol. 2017, 2017, doi: 10.1155/2017/3243570.

[202] M. Cheng, T. Qin, and J. Yang, “Node Localization Algorithm Based on Modified Archimedes Optimization Algorithm in Wireless Sensor Networks,” *J. Sensors*, vol. 2022, 2022, doi: 10.1155/2022/7026728.

[203] Z. Ansari, R. Ghazizadeh, and Z. Shokhmzan, “Gradient descent approach to secure localization for underwater wireless sensor networks,” *2016 24th Iran. Conf. Electr. Eng. ICEE 2016*, pp. 103–107, 2016, doi: 10.1109/IranianCEE.2016.7585498.

[204] F. A. Khan, A. H. Farooqi, and A. Derhab, “A comprehensive security analysis of LEACH++ clustering protocol for wireless sensor networks,” *J. Supercomput.*, vol. 75, no. 4, pp. 2221–2242, 2019, doi: 10.1007/s11227-018-2680-3.

[205] L. Mohammadpour, T. C. Ling, C. S. Liew, and A. Aryanfar, “A Mean Convolutional Layer for Intrusion Detection System,” *Secur. Commun. Networks*, vol. 2020, no. Ml, 2020, doi: 10.1155/2020/8891185.

[206] L. Xinlong and C. Zhibin, "DDoS Attack Detection by Hybrid Deep Learning Methodologies," *Secur. Commun. Networks*, vol. 2022, 2022, doi: 10.1155/2022/7866096.

[207] B. Hasan, S. Alani, and M. A. Saad, "Secured node detection technique based on artificial neural network for wireless sensor network," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 1, pp. 536–544, 2021, doi: 10.11591/ijece.v11i1.pp536-544.

[208] A. M. Pasikhani, J. A. Clark, and P. Gope, "Reinforcement-Learning-based IDS for 6LoWPAN," pp. 1049–1060, 2022, doi: 10.1109/trustcom53373.2021.00144.

[209] S. P. K. Gudla, S. K. Bhoi, S. R. Nayak, and A. Verma, "DI-ADS: A Deep Intelligent Distributed Denial of Service Attack Detection Scheme for Fog-Based IoT Applications," *Math. Probl. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/3747302.

[210] R. Khilar *et al.*, "Artificial Intelligence-Based Security Protocols to Resist Attacks in Internet of Things," *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022, doi: 10.1155/2022/1440538.

[211] S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, and K. Piamrat, "Federated Learning for intrusion detection system : Concepts , challenges and future directions," *Comput. Commun.*, vol. 195, no. September, pp. 346–361, 2022, doi: 10.1016/j.comcom.2022.09.012.

[212] O. Cheikhrouhou and A. Koubaa, "BlockLoc: Secure localization in the internet of things using blockchain," *2019 15th Int. Wirel. Commun. Mob. Comput. Conf. IWCMC 2019*, pp. 629–634, 2019, doi: 10.1109/IWCMC.2019.8766440.

[213] S. Algarni, F. Eassa, K. Almarhabi, A. Almalaise, and E. Albassam, "applied sciences Blockchain-Based Secured Access Control in an IoT System," pp. 1–16, 2021.

[214] M. P. Nath, S. N. Mohanty, and S. B. B. Priyadarshini, "Application of machine learning in wireless sensor network," *Proc. 2021 8th Int. Conf. Comput. Sustain. Glob. Dev. INDIACom 2021*, pp. 7–12, 2021, doi: 10.1109/INDIACom51348.2021.00003.

[215] Z. Cui *et al.*, "A Hybrid BlockChain-Based Identity Authentication Scheme for Multi-WSN," *IEEE Trans. Serv. Comput.*, vol. 13, no. 2, pp. 241–251, 2020, doi: 10.1109/TSC.2020.2964537.

[216] A. U. Khan and N. Javaid, "A blockchain scheme for authentication , data sharing and nonrepudiation to secure internet of wireless sensor things," *Cluster Comput.*, vol. 0123456789, 2022, doi: 10.1007/s10586-022-03722-z.

[217] R. Jeet, S. S. Kang, S. S. Hoque, and B. N. Dugbakie, "Secure Model for IoT Healthcare

System under Encrypted Blockchain Framework," vol. 2022, 2022.

[218] B. Ghimire and D. B. Rawat, "Recent Advances on Federated Learning for Cybersecurity and Cybersecurity for Federated Learning for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8229–8249, 2022, doi: 10.1109/JIOT.2022.3150363.

[219] A. Ahmed, S. Abdullah, M. Bukhsh, I. Ahmad, and Z. Mushtaq, "An Energy-Efficient Data Aggregation Mechanism for IoT Secured by Blockchain," *IEEE Access*, vol. 10, pp. 11404–11419, 2022, doi: 10.1109/ACCESS.2022.3146295.

[220] M. H. Ali, M. Fadlizolkipi, and A. Firdaus, "A hybrid Particle swarm optimization - Extreme Learning Machine approach for Intrusion Detection System," *2018 IEEE Student Conf. Res. Dev.*, pp. 1–4, 2018.

[221] C. Lyu, X. Zhang, Z. Liu, and C. H. Chi, "Selective Authentication Based Geographic Opportunistic Routing in Wireless Sensor Networks for Internet of Things Against DoS Attacks," *IEEE Access*, vol. 7, pp. 31068–31082, 2019, doi: 10.1109/ACCESS.2019.2902843.

[222] B. Kaur and D. Prashar, "Localization in Wireless Sensor Network: Techniques, Algorithms Analysis and Challenges," *2021 9th Int. Conf. Reliab. Infocom Technol. Optim. (Trends Futur. Dir. ICRITO 2021*, pp. 1–7, 2021, doi: 10.1109/ICRITO51393.2021.9596135.

[223] T. A. Alghamdi, I. Ali, N. Javaid, and M. Shafiq, "Secure Service Provisioning Scheme for Lightweight IoT Devices with a Fair Payment System and an Incentive Mechanism Based on Blockchain," *IEEE Access*, vol. 8, pp. 1048–1061, 2020, doi: 10.1109/ACCESS.2019.2961612.

[224] S. Salim, B. Turnbull, and S. Member, "A Blockchain-Enabled Explainable Federated Learning for Securing Internet-of-Things-Based Social Media 3 . 0 Networks," pp. 1–17, 2021.

[225] A. Rehman, S. Abbas, M. A. Khan, T. M. Ghazal, K. Muhammad, and A. Mosavi, "A secure healthcare 5 . 0 system based on blockchain technology entangled with federated learning technique," *Comput. Biol. Med.*, vol. 150, no. September, p. 106019, 2022, doi: 10.1016/j.compbiomed.2022.106019.

[226] Z. Mahmood and V. Jusas, "Blockchain-Enabled: Multi-Layered Security Federated Learning Platform for Preserving Data Privacy," *Electronics*, vol. 11, no. 10, p. 1624, 2022, doi: 10.3390/electronics11101624.

[227] T. Hassan, S. Aslam, and J. W. Jang, “Fully Automated Multi-Resolution Channels and Multithreaded Spectrum Allocation Protocol for IoT Based Sensor Nets,” *IEEE Access*, vol. 6, pp. 22545–22556, 2018, doi: 10.1109/ACCESS.2018.2829078.

[228] N. Nasser, Z. M. Fadlullah, M. M. Fouda, A. Ali, and M. Imran, “A lightweight federated learning based privacy preserving B5G pandemic response network using unmanned aerial vehicles: A proof-of-concept,” *Comput. Networks*, vol. 205, no. December 2021, p. 108672, 2022, doi: 10.1016/j.comnet.2021.108672.

[229] S. Roy, J. Li, and Y. Bai, “A Two-layer Fog-Cloud Intrusion Detection Model for IoT Networks,” *Internet of Things*, p. 100557, 2022, doi: 10.1016/j.iot.2022.100557.

[230] S. S. Mohar, S. Goyal, and R. Kaur, *Localization of sensor nodes in wireless sensor networks using bat optimization algorithm with enhanced exploration and exploitation characteristics*, vol. 78, no. 9. Springer US, 2022. doi: 10.1007/s11227-022-04320-x.

[231] A. Kumar, “A Hybrid Fuzzy System Based Cooperative Scalable and Secured Localization Scheme for Wireless Sensor Networks,” *Int. J. Wirel. Mob. Networks*, vol. 10, no. 3, pp. 51–68, 2018, doi: 10.5121/ijwmn.2018.10305.

[232] T. J. Nagalakshmi, A. K. Gnanasekar, G. Ramkumar, and A. Sabarivani, “Machine learning models to detect the blackhole attack in wireless adhoc network,” *Mater. Today Proc.*, vol. 47, no. xxxx, pp. 235–239, 2021, doi: 10.1016/j.matpr.2021.04.129.

[233] L. Liu, Z. Hu, and L. Wang, “Energy-efficient and privacy-preserving spatial range aggregation query processing in wireless sensor networks,” vol. 15, no. 7, 2019, doi: 10.1177/1550147719861005.

[234] W. Min, R. Chen, and S. He, “A secure data aggregation approach in hierarchical wireless sensor networks,” *ACM IMCOM 2016 Proc. 10th Int. Conf. Ubiquitous Inf. Manag. Commun.*, 2016, doi: 10.1145/2857546.2857637.

[235] N. Temene, C. Sergiou, C. Georgiou, and V. Vassiliou, “A Survey on Mobility in Wireless Sensor Networks,” *Ad Hoc Networks*, vol. 125, no. October, 2022, doi: 10.1016/j.adhoc.2021.102726.

[236] K. Nirmal Raja and M. Maraline Beno, “Secure Data Aggregation in Wireless Sensor Network-Fujisaki Okamoto(FO) Authentication Scheme against Sybil Attack,” *J. Med. Syst.*, vol. 41, no. 7, pp. 2–7, 2017, doi: 10.1007/s10916-017-0743-2.

[237] D. Vinodha and E. A. Mary Anita, “Secure Data Aggregation Techniques for Wireless Sensor Networks: A Review,” *Arch. Comput. Methods Eng.*, vol. 26, no. 4, pp. 1007–1027,

Sep. 2019, doi: 10.1007/s11831-018-9267-2.

[238] J. Yun, S. Seo, and J. M. Chung, “Centralized Trust-Based Secure Routing in Wireless Networks,” *IEEE Wirel. Commun. Lett.*, vol. 7, no. 6, pp. 1066–1069, 2018, doi: 10.1109/LWC.2018.2858231.

[239] R. Rastogi, S. Srivastava, Tarun, M. Singh Manshahia, Varsha, and N. Kumar, “A hybrid optimization approach using PSO and ant colony in wireless sensor network,” *Mater. Today Proc.*, no. xxxx, 2021, doi: 10.1016/j.matpr.2021.01.874.

[240] Y. Zhou, N. Wang, and W. Xiang, “Clustering Hierarchy Protocol in Wireless Sensor Networks Using an Improved PSO Algorithm,” *IEEE Access*, vol. 5, pp. 2241–2253, 2017, doi: 10.1109/ACCESS.2016.2633826.

[241] J. Cui, K. Boussetta, and F. Valois, “Classification of data aggregation functions in wireless sensor networks,” *Comput. Networks*, vol. 178, no. May, p. 107342, 2020, doi: 10.1016/j.comnet.2020.107342.

[242] S. Kaur and R. Mahajan, “Hybrid meta-heuristic optimization based energy efficient protocol for wireless sensor networks,” *Egypt. Informatics J.*, vol. 19, no. 3, pp. 145–150, 2018, doi: 10.1016/j.eij.2018.01.002.

[243] A. Jain and A. K. Goel, “Energy Efficient Fuzzy Routing Protocol for Wireless Sensor Networks,” *Wirel. Pers. Commun.*, vol. 110, no. 3, pp. 1459–1474, 2020, doi: 10.1007/s11277-019-06795-z.

[244] H. W. Kim, K. Lee, C. Moon, and Y. Nam, “Comparative Analysis of Machine Learning Algorithms along with Classifiers for AF Detection using a Scale,” *1st Int. Conf. Artif. Intell. Inf. Commun. ICAIIC 2019*, no. May, pp. 427–429, 2019, doi: 10.1109/ICAIIC.2019.8669084.

[245] S. Kalmegh, “Analysis of WEKA Data Mining Algorithm REPTree , Simple Cart and RandomTree for Classification of Indian News,” *Int. J. Innov. Sci. Eng. Technol.*, vol. 2, no. 2, pp. 438–446, 2015.

[246] S. Jukic, M. Saracevic, A. Subasi, and J. Kevric, “Comparison of ensemble machine learning methods for automated classification of focal and non-focal epileptic EEG signals,” *Mathematics*, vol. 8, no. 9, 2020, doi: 10.3390/math8091481.

[247] V. D. Katkar and S. V. Kulkarni, “Experiments on detection of Denial of Service attacks using ensemble of classifiers,” *Proc. 2013 Int. Conf. Green Comput. Commun. Conserv. Energy, ICGCE 2013*, pp. 837–842, 2013, doi: 10.1109/ICGCE.2013.6823550.

[248] M. Premkumar, T. Vinay, P. Sundararajan, and G. Mohanbabu, “Dynamic Defense Mechanism for DoS Attacks in Wireless Environments Using Hybrid Intrusion Detection System and Statistical Approaches,” vol. 3651, pp. 965–970, 1848.

[249] M. Maheswari and R. A. Karthika, “A Novel QoS Based Secure Unequal Clustering Protocol with Intrusion Detection System in Wireless Sensor Networks,” *Wirel. Pers. Commun.*, no. 0123456789, 2021, doi: 10.1007/s11277-021-08101-2.

[250] M. Agarwal, D. Pasumarthi, S. Biswas, and S. Nandi, “Machine learning approach for detection of flooding DoS attacks in 802.11 networks and attacker localization,” *Int. J. Mach. Learn. Cybern.*, vol. 7, no. 6, pp. 1035–1051, 2016, doi: 10.1007/s13042-014-0309-2.

[251] N. N. Gana and S. M. Abdulhamid, “Machine Learning Classification Algorithms for Phishing Detection: A Comparative Appraisal and Analysis,” *2019 2nd Int. Conf. IEEE Niger. Comput. Chapter, Niger. 2019*, 2019, doi: 10.1109/NigeriaComputConf45974.2019.8949632.

[252] J. Ramprasath and V. Seethalakshmi, “Mitigation of Malicious Flooding in Software Defined Networks Using Dynamic Access Control List,” *Wirel. Pers. Commun.*, no. 0123456789, 2021, doi: 10.1007/s11277-021-08626-6.

[253] M. F. Suleiman and B. Issac, “Performance Comparison of Intrusion Detection Machine Learning Classifiers on Benchmark and New Datasets,” pp. 19–23, 2021, doi: 10.1109/iccta45985.2018.9499140.

[254] N. A. Awad, “Enhancing network intrusion detection model using machine learning algorithms,” *Comput. Mater. Contin.*, vol. 67, no. 1, pp. 979–990, 2021, doi: 10.32604/cmc.2021.014307.

Appendix A: List of publications

a. Research Papers in International Journal

1. Gebrekiros Gebreyesus Gebremariam, J. Panda, S. Indu, "Localization and Detection of Multiple Attacks in Wireless Sensor Networks Using Artificial Neural Network", Wireless Communications and Mobile Computing, vol. 2023, Article ID 2744706, 29 pages, 2023. **SCIE**, (IF: 2.146), (**Accepted & Published**), https://doi.org/10.1155/2023/2744706
2. Gebrekiros Gebreyesus Gebremariam, J. Panda, S. Indu, "Blockchain-Based Secure Localization against Malicious Nodes in IoT-Based Wireless Sensor Networks Using Federated Learning", Wireless Communications and Mobile Computing, vol. 2023, Article ID 8068038, 27 pages, 2023. **SCIE**, (**IF: 2.146**), (**Accepted & Published**), https://doi.org/10.1155/2023/8068038
3. G. G. Gebremariam, J. Panda, and S. Indu, "**Design of Advanced Intrusion Detection Systems Based on Hybrid Machine Learning Techniques in Hierarchically Wireless Sensor Networks**," 2023, doi: 10.1080/09540091.2023.2246703**.,** Connection science, **(Accepted and published)**
4. G. G. Gebremariam, J. Panda, and S. Indu "**Clustering and Aggregation for Secure Wireless Sensor Networks Using Machine Learning Classifiers**". Telecommunication systems, SCIE, (IF: 2.336), (Communicated with minor revision).
5. G. G. Gebremariam, J. Panda, and S. Indu "**Secure Localization Techniques in Wireless Sensor Networks against Routing Attacks Based on Hybrid Machine Learning Models**". Alexandria Engineering Journal, (Communicated with minor revision).
6. G. G. Gebremariam, J. Panda, and S. Indu, "**Detection and Analysis of Flooding Attacks in Wireless Sensor Networks**". Wireless Personal Communications, SCIE, (IF: 2.017), (Under Review).

b. Research Papers in International Conference

1. G. G. Gebremariam, J. Panda and S. Indu, "Secure Intrusion Detection System for Hierarchically Distributed Wireless Sensor Networks," 2021 International Conference on Industrial Electronics Research and Applications (ICIERA), New Delhi, India, 2021, pp. 1-6, (Accepted and published), doi: 10.1109/ICIERA53202.2021.9726753.
2. Gebremariam, G.G., Panda, J., Indu, S. (2023), "Secure Localization Techniques in Wireless Sensor Networks Against Routing Attacks Using Machine Learning Models", Soft Computing and Signal Processing. ICSCSP 2022. Smart Innovation, Systems and Technologies, vol 313. Springer, Singapore. https://doi.org/10.1007/978-981-19-8669-7_52
3. G. G. Gebremariam, J. Panda and S. Indu. "Security in Wireless Sensor Networks Against Routing Attacks Based On Secure Clustering and Routing," 2023 International Conference on SOLI.

Biodata

Gebrekiros Gebreyesus Gebremariam is an individual with a strong educational background and significant involvement in research and community service. He obtained his Bachelor of Science degree in electrical and computer engineering from **Mekelle University in Tigray, Ethiopia** in 2015. Following that, he pursued his Master of Technology degree in Electronics and Communication Engineering from **Punjabi University in Patiala, Punjab, India**, which he completed in 2018.

Currently, Gebrekiros holds the position of a lecturer at **Raya University in Tigray, Ethiopia**. He is actively engaged in teaching and contributing to the academic development of students. Additionally, he has participated in various community service and research training activities, which are part of a capacity building program.

In 2019, Gebrekiros embarked on a Ph.D. program as a full-time student in the Electronics and Communication department at **Delhi Technological University in New Delhi, India**. His research advisors are Professors **J. Panda** and **S. Indu**. His current doctoral research is focused on ensuring the safety of wireless sensor networks integrated into Internet of Things (IoT)-based artificial intelligence. Within this research domain, Gebrekiros has proposed several rigorous and effective methodologies aimed at enhancing the security and reliability of such networks.

Gebrekiros Gebreyesus Gebremariam's academic journey and research contributions reflect his dedication to advancing knowledge in the field of electronics and communication engineering, particularly in the context of wireless sensor networks and IoT-based on artificial intelligence.