ANOMALY-BASED NETWORK INTRUSION DETECTION SYSTEM USING MACHINE LEARNING ALGORITHMS

A Thesis Submitted
In Partial Fulfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

by

Sumedha Seniaray
(2K19/PHDCO/16)

Under the Supervision of

Prof. Rajni Jindal
Department of Computer Science & Engineering

Department of Computer Science & Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahabad Daulatpur, Main Bawana Road, Delhi-110042, INDIA

October, 2025

©DELHI TECHNOLOGICAL UNIVERSITY-2025
ALL RIGHTS RESERVED

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahabad Daulatpur, Main Bawana Road, Delhi-110042, INDIA

CANDIDATE'S DECLARATION

I, **Sumedha Seniaray (2K19/PHDCO/16)** hereby certify that the work which is being presented in the thesis entitled **"Anomaly-based Network Intrusion Detection System using Machine Learning Algorithms"** in partial fulfillment of the requirements for the award of the Degree of Doctor of Philosophy, submitted in the Department of Computer Science & Engineering, Delhi Technological University is an authentic record of my own work carried out during the period from 2019 to 2025 under the supervision of Prof. Rajni Jindal.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Sumedha Seniaray
(2K19/PHDCO/16)

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Prof. Rajni Jindal
Professor, Department of Computer Science & Engineering, DTU

Supervisor

Prof. Arun Sharma
Professor, Department of Information Technology and Dean (Academic Affairs), IGDTUW

External Examiner

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahabad Daulatpur, Main Bawana Road, Delhi-110042, INDIA

CERTIFICATE

This is to certify that **Ms. Sumedha Seniaray (2K19/PHDCO/16)** has carried out her research work presented in this thesis entitled **"Anomaly-based Network Intrusion Detection System using Machine Learning Algorithms"** for the award of **Doctor of Philosophy** from Department of Computer Science & Engineering, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work, and studies are carried out by the student herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/ Institution to the best of my knowledge.

Prof. Rajni Jindal
(Supervisor)
Department of Computer Science & Engineering
Delhi Technological University
Delhi

Date: 08/10/2025
Place: Delhi

ACKNOWLEDGMENTS

It is with immense gratitude and deep appreciation that I acknowledge the support and encouragement of all those who have contributed to the successful completion of this thesis. I would like to express my profound gratitude to the Hon'ble Vice Chancellor, DTU, Prof. Prateek Sharma sir, for fostering an environment that encourages research, innovation, and academic excellence.

I extend my heartfelt gratitude to my supervisor, Prof. Rajni Jindal, whose expert guidance, consistent support, and insightful suggestions have been instrumental in shaping this research. Her mentorship has not only enhanced my academic skills but also instilled in me a sense of discipline and perseverance that I will carry forward in all future endeavours.

I am equally grateful to Prof. Ramesh Srivastava, Head of the Department of Applied Mathematics, for his constant encouragement, positive outlook, and inspiring leadership, which created an environment that was conducive to focused research and learning. I would also like to thank all the faculty members of the Department of Computer Science & Engineering and the Department of Applied Mathematics for their valuable inputs, encouragement, and support at various stages of my research work. Their guidance and feedback helped me navigate challenges with clarity and confidence.

I am deeply indebted to my mother and father, whose unconditional love, blessings, and faith in me have been my greatest strength. Their constant moral support and encouragement have sustained me through the highs and lows of this journey. I would also like to thank my husband, Jaspreet, whose understanding and encouragement have meant a great deal to me, and whose patience and thoughtful care created a peaceful and nurturing environment that allowed me to stay focused and complete my thesis with clarity and determination. I sincerely thank my in-laws for their affection, understanding, and quiet strength that have supported me throughout. A special thanks to my brother, Sahil, for always believing in me, motivating me with his words and actions, and being a constant source of positivity and encouragement.

I would also like to extend my warm thanks to my dear friends and colleagues, Trasha, Anshul, Payal, Ravindra Sir, and Anjana Ma'am. Your friendship, encouragement, and thoughtful discussions have greatly enriched my academic experience.

Whether it was sharing ideas, offering emotional support, or simply being there, your presence made this journey more fulfilling.

Finally, I would like to thank everyone who, directly or indirectly, contributed to the successful completion of this thesis. I couldn't have reached this milestone without your support, and I'm sincerely grateful for the invaluable role you've played in my life.

Date: 08/10/2025 **(Sumedha Seniaray)**
Place: Delhi

ABSTRACT

The rapid advancement of technology not only simplifies life but also introduces numerous security challenges. Over the years, as the Internet has evolved, the frequency and sophistication of cyberattacks have increased significantly, targeting individuals, organizations, and critical infrastructures. This growing threat underscores the vital need for robust security frameworks. Intrusion Detection Systems (IDS) play a crucial role in continuously monitoring network activity, identifying malicious behaviours, and mitigating potential attacks in real time. Hence, anomaly-based network intrusion detection powered with machine learning techniques is proposed in this thesis to develop intelligent and adaptive IDS solutions, which are crucial for maintaining strong cybersecurity defences.

This thesis aims to enhance the effectiveness of Intrusion Detection Systems by initially addressing the challenge of selecting the most relevant features from high-dimensional network traffic data. The presence of redundant and irrelevant features can lead to increased computational complexity and reduced detection accuracy. We proposed a three-phase network-based IDS to counter this issue, where we developed a dynamic mutual information-based genetic algorithm (DMI-GA), a novel feature selection technique designed to identify an optimal set of features. By integrating mutual information to measure feature relevance and a genetic algorithm to optimize selection, DMI-GA enhances both the efficiency and accuracy of IDS models. Unlike many existing feature selection methods that evaluate each feature independently and fail to account for feature dependencies, our approach considers the relationships between features, leading to more informed selection and improved computational performance. This method not only reduces dimensionality but also ensures that the most significant features contribute to better attack detection.

The high dimensionality of data degrades IDS performance, causing sparsity issues that obscure meaningful patterns. It also increases the risk of overfitting, making models learn noise instead of actual attack behaviours, reducing their ability to detect new threats. Also, since dataset quality is crucial for accurately detecting and classifying intrusions, the presence of highly imbalanced data, where benign network packets significantly outnumber anomalous ones, can deteriorate classification performance.

Thus, we developed another anomaly-based IDS in conjunction with machine learning techniques and a novel modified picture fuzzy clustering-based approach, $mP_{ic}FC$, on the dimensionality-reduced dataset. This approach incorporates an additional decision-making layer to handle uncertainty more effectively. By differentiating between partial membership and complete non-membership, it enables more precise classifications. The inclusion of refusal or hesitation degrees helps minimize bias in clustering, preventing uncertain data points from disproportionately influencing the results. Moreover, the proposed framework addresses the class imbalance by reducing bias toward the majority class, using the Synthetic Minority Oversampling Technique (SMOTE), which ultimately improved the model's accuracy.

To address the growing need for real-time threat detection, we propose HIL-IDS, a real-time Intrusion Detection System based on a hybrid incremental learning approach. HIL-IDS continuously monitors network traffic, detects anomalies, and adapts to evolving cyber threats with minimal latency. It integrates the Hoeffding Tree for incremental supervised learning, leveraging its efficiency in processing streaming data, and an ensemble of Isolation Forest and K-Means for unsupervised anomaly detection, effective in identifying novel attack patterns without prior labels. Confidence scores from the combination of these supervised and unsupervised models are evaluated to enhance the interpretability of the proposed framework. To maintain robustness against shifting data distributions, drift detection enables adaptation to emerging threats in real time. By combining multiple anomaly detection methods in an ensemble, HIL-IDS improves the likelihood of detecting diverse attack types. While hybrid intrusion detection systems exist, the integration of incremental learning in both supervised and unsupervised components allows HIL-IDS to dynamically adapt to evolving attacks and network traffic in real time, making it highly suitable for modern, dynamic network environments.

The performance of the developed methods is compared with the various contemporary models on the sophisticated network traffic datasets using quantitative and statistical sampling assessments. The empirical results, along with statistical tests, show the superiority of the proposed methods over the existing methods for intrusion detection. Overall, the findings establish a strong foundation for future research in developing more adaptive and intelligent intrusion detection systems, enhancing real-time threat detection, and improving the scalability and efficiency of cybersecurity solutions.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the rapid advancement of technology and the increasing reliance on computer networks, cybersecurity has become a critical concern. Organizations, governments, and individuals are constantly at risk of cyber threats. Thus, the growing sophistication of cyber-attacks necessitates the development of robust security mechanisms to protect network infrastructures.

This chapter begins with an overview of network security, highlighting its associated challenges. It then introduces the Intrusion Detection System (IDS) and discusses its types, the network attacks it addresses, the motivation behind the research, the problem statement, and the contributions made. The final section presents the thesis structure.

1.1 Background

The swift development of networks witnesses trillions of data transfers every day. The sheer magnitude of this data transfer provides opportunities for intruders to develop novel and unconventional techniques to infiltrate and exploit it. Cyber threat is one of the challenging fields in this current data-savvy world. Thus, robust network privacy and security measures are required to minimize the chances of data leaks and mitigate potential network security threats. While various protection methods are available, such as firewalls and anti-viruses, it is essential to acknowledge that these methods are not infallible. They are susceptible to failure when attackers introduce unknown malware or an extensive amount of it into the system. The growth of malware infection has

immensely increased from 12.4 million to 812.67 million between 2009 and 2018 [1]. Given the continuous escalating intensity and variety of cyber-attacks, developing efficient and effective techniques to counteract these threats becomes crucial.

An attack or intrusion is a series of actions aimed at compromising the integrity, privacy, or availability of a service within a computer environment. The first intrusion detection system was proposed in 1987 [2]. Since then, many researchers have developed numerous intrusion detection models. Intrusion detection encompasses the systematic process of identifying and analyzing various events that occur within a system or network and responding quickly to any malicious activities detected to reduce their impact and ensure system security [3]. An Intrusion Detection System helps identify various forms of malicious or abnormal network traffic and computer activity that may be difficult to detect using a conventional firewall or may be unknown to the user [4]. This includes network attacks targeting vulnerable services, data-driven attacks on applications, host-based attacks such as unauthorized system or software logins, privilege escalation, access to sensitive user files and data, and malware.

Intrusion detection systems and firewalls are both integral components of network security, but they serve distinct purposes. A firewall acts as a protective barrier, monitoring external traffic to prevent intrusions before they occur. It restricts access between networks to block unauthorized connections. However, if an attack originates within the network, the firewall does not generate an alert. In contrast, an IDS identifies suspicious activity after an intrusion has occurred and triggers an alarm to notify administrators. While firewalls protect against external threats by blocking unauthorized access attempts, IDS monitors network activity and alerts the system upon detecting malicious behaviour. The primary objective of an IDS is to generate an alert when an attack or network intrusion takes place [5, 6]. Various application areas of intrusion detection include the Internet of Things (IoT), smart cities, big data environments, wireless sensor networks (WSNs), and more [7, 8, 9].

In terms of intrusion detection methods, IDS can be broadly categorized as:

- **Misuse or Signature-based IDS:**
  This approach detects attacks by identifying specific "signature" patterns. For instance, malware may contain known malicious instructions or byte sequences within network traffic. A signature-based IDS is effective in detecting known at-

tacks; however, it struggles to identify new or previously unknown attack patterns. Consequently, its effectiveness depends on how frequently its database is updated over time.

- **Anomaly-based IDS:**
  In contrast to signature-based IDS, anomaly-based IDS detect unknown attacks that are difficult to identify using signature-based methods. These systems compare known behaviours with incoming behaviours using machine learning algorithms. As a result, any abnormal or unusual activity is flagged. Therefore, anomaly-based IDS are effective in detecting both known and unknown attacks within a network.

Based on the features utilized or the source data for intrusion detection, IDS can be categorized into two types:

- **Host-based IDS (HIDS):**
  HIDS can detect intrusions on the host system by inspecting the data such as server logs, application system audits, database records, etc. or by identifying if any unauthorised access has been made. It monitors the dynamic behavior of individual components or the system as a whole, as well as the system's state based on its configuration. HIDS is particularly effective in inspecting a specific host targeted by network packets and tracking resource access by different programs. For instance, it can detect if a word processor unintentionally modifies the system password database. Therefore, HIDS plays a crucial role in analyzing and enforcing a system's security policies [10].

- **Network-based IDS (NIDS):**
  NIDS helps detect threats targeting a computer network, such as malicious hacking activities and denial-of-service (DoS) attacks. NIDS detects malicious data by monitoring network traffic and examining if there is any unusual activity on the network. It examines both network traffic and the data transmitted between systems within the network. Before the intrusion spreads to the systems present in the network, an alarm is initiated to notify administrators about the threat that has been encountered.

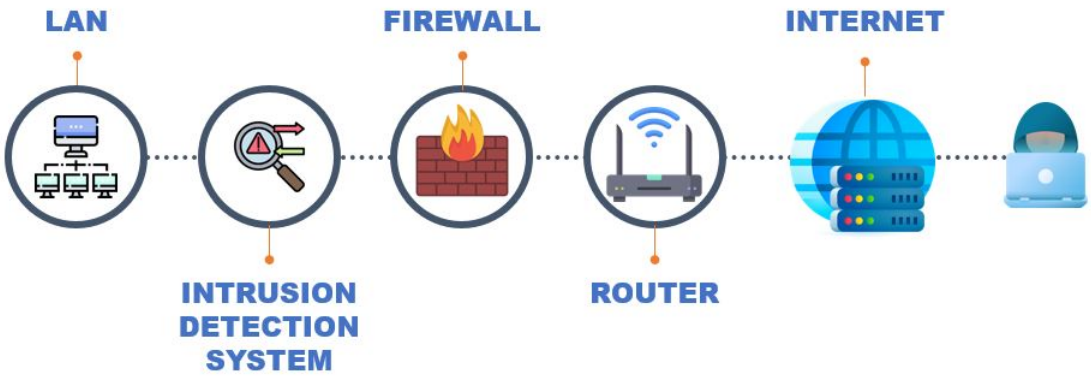

Fig. 1.1: Generic Network-based Intrusion Detection System

Fig. 1.1 presents the working of a generic network-based IDS, which screens the traffic flowing through the network under inspection to detect any unusual activity. Incoming and outgoing network traffic is routed through the central router, which manages the data packet distribution within the internal network and to external destinations. The firewall, positioned between the Internet and the internal network, acts as the first line of defence. It filters traffic based on pre-configured security rules, such as IP address restrictions, port blocking, or protocol validation. However, it mainly uses static filtering rules and may not detect complex or sophisticated attacks, which could bypass simple filtering mechanisms. After passing through the firewall, the network traffic reaches the IDS. The IDS continuously monitors this traffic, looking for patterns or behaviours indicative of attacks, such as unusual access requests, spikes in data volume, or unauthorized use of protocols. When the IDS identifies traffic that matches its predefined patterns or deviates significantly from normal behaviour, it triggers an alert for the network administrator. Simultaneously, it logs these events for further analysis, which helps in identifying attack trends and strengthening future defences.

In this study, we focus on developing an anomaly-based network intrusion detection system. Monitoring network traffic flows is essential to detect not only known threats but also unknown or abnormal traffic patterns. An anomaly-based intrusion detection system continuously analyzes network traffic, comparing it to normal traffic behaviour. If it identifies unusual patterns or anomalies, it triggers an alarm, signalling a potential threat. Based on this comparison, network traffic is classified as either benign (normal) or malicious (abnormal).

1.1.1 Benchmark IDS Datasets and Types of Network Attacks

To develop and evaluate intrusion detection systems, researchers rely on various publicly available datasets. These datasets play a crucial role in cybersecurity by providing real-world network traffic data, containing labelled instances of both normal and malicious network traffic, allowing for the training and testing of machine learning models. Table 1.1 presents some widely used benchmark NIDS datasets [11, 12, 13, 14, 15, 16, 17, 18] with their respective description.

Table 1.1: Summary of Benchmark Network Intrusion Detection System Datasets

Dataset Name	Number of Features	Attack Types	Description
KDD Cup'99	41	DoS, R2L, U2R, Probe	Derived from the 1998 DARPA Intrusion Detection Evaluation Program, where network traffic was collected over a simulated environment and preprocessed into connection-based records. It is labeled data with 4,898,431 data points but with high redundancy.
NSL-KDD	41	DoS, R2L, U2R, Probe	Enhanced version of KDD Cup'99 with no redundant instances in 125,973 training and 22,544 testing dataset instances.
CAIDA	15	DDoS	This unlabelled dataset includes around one hour of anonymized network traffic traces captured during a DDoS attack.
Kyoto 2006+	24	Normal and Attack classes	Developed on 3 years of honeypots and Darknet sensors by Kyoto University with no manual network data labelling.
ISCX 2012	20	DoS, DDoS, Brute Force, Infiltration	Developed by the University of New Brunswick, labeled network traffic data captured over seven days.
UNSW-NB15	49	DoS, Fuzzers, Backdoors, Analysis, Exploits, Generic, Shell Code, Reconnaissance, Worms	Generated by the Australian Centre for Cyber Security (ACCS), with 49 features and 2,540,044 instances, extracted from network traffic, divided into five categories: Flow, Basic, Content, Time features, and Additional generated features.
CIC-IDS2017	80	DoS, DDoS, Brute Force, Heartbleed, Botnet, Infiltration, Web attack	Realistic network traffic, generated by Canadian Institute for Cybersecurity (CIC) over 5 days using network profiles, covering real-world traffic and user behaviours.
CSE-CIC-IDS2018	80	DoS, DDoS, Brute Force, Botnet, Infiltration, Wen attack	Developed in 2018 by a collaboration between the Canadian Institute for Cybersecurity (CIC) and the Communications Security Establishment (CSE). It was prepared in an infrastructure with 50 attacking machines and a victim organization with 420 systems and 30 servers in a time span of 10 days, resulting in capturing 16,233,002 network traffic instances.
CIC-DDoS2019	78	DDoS	A labelled dataset created by the Canadian Institute for Cybersecurity for detecting Distributed Denial of Service (DDoS) attacks. It includes realistic benign and attack traffic generated in a controlled environment, covering various DDoS types.

The diversity of attack types included in these datasets helps assess IDS performance against different cyber threats for network security. Understanding these attack categories is essential for developing robust anomaly-based IDS models. Some of the common types of network attack found in the IDS datasets are described in Table 1.2:

Table 1.2: Description of common Network Traffic Attacks in IDS datasets

Network Traffic Attack	**Description**
Denial-of-Service (DoS)	The attack occurs when a system targets the victim system or server. It overloads network resources to disrupt services.
Distributed Denial-of-Service (DDoS)	The attack occurs when multiple devices target a victim system, server, or network, and overwhelm it with excessive traffic.
Probe attacks	Probe attacks are also known as scanning attacks, used by attackers to gather information about a network, such as active hosts, open ports, and running services for future exploitation.
User-to-Root (U2R)	It occurs when an attacker gains unauthorized root (administrator) privileges on a system by exploiting local vulnerabilities to gain full control over the system.
Remote-to-Local (R2L)	It occurs when an attacker, operating remotely, gains unauthorized access to a local machine by sending malicious packets or requests to it.
Brute Force	The attackers try trial-and-error hacking techniques to crack login credentials or passwords.
Heartbleed	It is a severe vulnerability in the OpenSSL cryptographic library, which allows attackers to steal sensitive data from the memory of vulnerable servers.
Botnet	It allows multiple attackers to control and penetrate the security of multiple victim devices and remotely command and organize them, used for malicious activities like spam, DDoS, or fraud.
Infiltration	This attack gains unauthorized access within the networked system via malware or exploitation, where the infiltrator uses various techniques to compromise or breach the system.
Web attack	The attacker gains unauthorized access, retrieves sensitive information by targeting the vulnerabilities in trusted websites by injecting malware, for instance, SQLInjection.

These datasets and attack types provide a foundation for the development and evaluation of IDS models, ensuring their effectiveness in real-world cybersecurity scenarios.

1.1.2 Machine Learning (ML) Algorithms

Machine learning algorithms play a crucial role in anomaly-based network intrusion detection systems by enabling the automated identification of malicious activities based on patterns and statistical deviations from normal network behavior. Unlike signature-

based detection methods, ML-based approaches can generalize from previously unseen attacks, making them more effective against novel threats. This subsection explores various machine learning algorithms used for network intrusion detection, their advantages, and their limitations.

Machine learning techniques can be broadly classified into two categories, supervised and unsupervised ML techniques. Supervised learning relies on labeled datasets where each data instance is explicitly categorized as either normal or malicious. This allows models to learn decision boundaries based on known attack patterns and classify future network traffic accordingly. Supervised ML can be used for various tasks, including binary classification, multiclass classification, integrating the predictions of multiple machine learning models to enhance accuracy, and estimating continuous values by analyzing patterns and relationships within the data. Supervised learning excels in detecting known attack types with high accuracy but may sometimes struggle with unseen threats due to its dependence on labeled data.

Unsupervised learning, in contrast, does not require labeled data. Instead, it identifies anomalies by detecting deviations from normal traffic patterns and groups data points into attack subsets. Unsupervised learning is particularly effective in detecting novel attacks that do not have predefined signatures. However, it may often generate false positive rates because normal variations in traffic can be mistakenly classified as anomalies.

Some of the widely used state-of-the-art machine learning models for intrusion detection, including those utilized in this study, are outlined below:

- **Logistic Regression (LR):** LR [19] is a supervised ML classifier and a non-linear regression model. LR helps in both multiclass and binary classification. Data fitting into the logistic function predicts the occurrence of an event's probability, and the function's value ranges from 0 to 1. This function helps in mapping predictions and probabilities. The standard Logistic function called $sigmoid$ is presented by Eqn. (1.1).

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{1.1}$$

- **Decision Tree (DT):** The decision tree is a supervised tree-structured classifier that helps not only in classification but also in regression [20, 21]. DT separates

the dataset into multiple homogeneous sets on the basis of independent variables and/ or significant attributes in order to make as discrete groups as possible. The tree's nodes represent some event, and the branches or edges represent the decision rules or conditions that categorize the data into separate groups.

- **Naïve Bayes (NB):** A supervised ML classification technique based on Bayes' theorem with "naïve" postulation of independence between attributes, that is, the presence of a particular feature within a class is disparate from any other feature's presence [22]. Prediction is made by evaluating instance probabilities of every class and then choosing the maximum probability class value, as described by Eqn. (1.2).

$$P(X|Y) = \frac{P(Y|X) * P(X)}{P(Y)}. \tag{1.2}$$

  where, $P(X|Y)$ is the probability of the target class with the given predictor attribute, $P(X)$ is the probability of the class, $P(Y)$ is the probability of the predictor, and $P(Y|X)$ is the probability of the predictor attribute with the given class.

- ***k*-Nearest neighbour (*k*-NN):** The *k*-Nearest Neighbour rule was first introduced in 1951 by Fix and Hodges [23] and further expanded by [24]. *k*-NN is one of the simplest supervised ML classifiers, and it can also be used for regression. This algorithm makes use of proximity to perform predictions or classification for grouping an individual data point. The predicted label for the input data point is determined by this classifier by taking the class label that is most common among the *k* neighbours.

- **Random Forest (RF):** RF is a complex non-linear supervised ensemble learning technique used for regression as well as classification [25]. It is a collection or forest of multiple decision trees. The outcomes of all the sets of decision trees together result in the RF's prediction result. In the model, the higher the number of decision trees, the higher is the Random Forest's prediction accuracy, which would not over-fit the model.

- **Support Vector Machine (SVM):** This supervised binary classification ML technique aims to assess the hyperplane by separating the dataset into two distinct classes, maximising the margin amongst all the attack classes [26, 27]. Support vectors, developed by segregating the training dataset into several subsets, are used to estimate predictions. It is dependent on the type of parameters and the kernels employed. SVM can also perform multiclass classification in a custom cascading manner.

- **K-Means:** K-Means is an unsupervised clustering ML technique that classifies the unlabelled data into K clusters [28]. Based on some similarities between the data points, the clusters are formulated. K-Means selects k number of points for each cluster, known as centroids. The measure of similarity between these data points is evaluated based on the distance between them. Data points within a cluster are homogeneous in nature, whereas the clusters are heterogeneous from one another.

Machine learning algorithms offer a powerful framework for anomaly-based network intrusion detection. By leveraging supervised, unsupervised, and ensemble models, ML-based NIDS can identify various types of intrusions. Machine learning provides several benefits in this context, including the ability to detect unknown or novel attacks, process large-scale network traffic data in real-time, and adapt to evolving cyber threats and changing network environments.

1.2 Motivation

Cybersecurity has become an ever-growing challenge due to the increasing volume and sophistication of cyber threats. Existing intrusion detection systems often fail to provide comprehensive protection against evolving attack techniques, emphasizing the need for more intelligent and adaptive detection mechanisms. This research is motivated by the growing complexity and frequency of cyber attacks, which pose significant risks to individuals and organizations. Traditional IDS solutions, particularly signature-based methods, struggle to keep pace with the evolving nature of cyber threats, whereas anomaly-based IDS offer a more proactive approach.

The increasing sophistication of cyber threats has exposed critical limitations in traditional machine learning-based IDS. Many existing IDS struggle with high-dimensional data, leading to elevated false alarm rates and computational inefficiencies. Additionally, conventional models often rely on outdated datasets, limiting their ability to detect emerging and rare attack patterns. Most traditional approaches employ either supervised or unsupervised learning on lower-dimensional data, restricting their effectiveness in handling complex network environments. Furthermore, commonly used feature selection methods often overlook feature dependencies, resulting in suboptimal classification performance. Another major challenge is that most IDS approaches rely solely on supervised learning, which requires labeled data and may not effectively address evolving, unlabeled threats. As real-time IDS solutions become increasingly essential, traditional ML-based IDS, trained on static datasets, struggle to adapt to dynamic network environments where data distributions change over time. These challenges highlight the need for more robust and adaptive approaches to anomaly-based intrusion detection. This research aims to address these critical gaps, ultimately strengthening network security, protecting sensitive information, and mitigating the risks associated with intrusions.

1.3 Problem Statement

Traditional IDS often struggle to detect previously unseen attacks due to its reliance on predefined signatures. While anomaly-based IDS can identify novel attacks, they tend to produce high false positive rates, reducing their reliability in real-world scenarios. The effectiveness of an anomaly-based IDS depends on its ability to extract meaningful features and accurately classify network traffic patterns.

Existing supervised and unsupervised machine learning models for anomaly detection face several challenges, including high false alarm rates, inefficient feature selection, and computational overhead when processing large-scale network traffic data. Class imbalance further affects traditional machine learning-based IDS methods, leading to biased detection that favors majority classes while underrepresenting minority attack instances. Additionally, optimizing feature selection and dimensionality reduction is crucial to improving classification accuracy while minimizing information loss. Many existing approaches fail to address the high dimensionality of network traffic data effectively, resulting in performance bottlenecks, particularly in real-time environments.

Datasets such as CIC-IDS2017 and CIC-IDS2018 provide rich, real-world network traffic data, but processing and analyzing them efficiently remains a challenge due to their complexity and size. Moreover, real-time network traffic analysis introduces additional complexities that require scalable solutions for efficient processing of streaming data.

This research aims to address these limitations by proposing a framework to enhance the intrusion detection process, apply dimensionality reduction techniques before classification, and accurately cluster network traffic patterns. The effectiveness of an anomaly-based IDS also largely depends on its ability to extract meaningful features. Therefore, the study proposes to optimize the feature selection process for IDS and leverage machine learning models to enhance detection accuracy.

Furthermore, real-time network data collection and processing will be explored to improve IDS adaptability in dynamic network environments. By addressing these challenges, this study seeks to contribute to the development of a more robust and efficient IDS framework capable of detecting diverse cyber threats with higher accuracy and lower computational overhead.

1.4 Main Contributions of the Thesis

The main contributions of this research include the design and development of frameworks to classify network traffic flows as benign (normal) or malicious attacks, as well as detecting various types of network attacks. Furthermore, this research aims to develop an IDS capable of detecting intrusions in a dynamic network environment.

1.4.1 Anomaly-based NIDS using Feature Selection

Various IDS frameworks for detecting intrusions are available in the literature. However, limited research has focused on intrusion detection using supervised and unsupervised machine learning models with feature selection. A novel feature selection model is developed to enhance the performance efficiency of the ML algorithms.

- We developed a hybrid feature selection technique based on mutual information and an evolutionary genetic algorithm. This approach considers not only indi-

vidual features but also their degree of association with each other and the target variable to obtain an optimal feature set.

- A dynamic fitness function, serving as the core of the genetic algorithm, is developed to identify optimum features.
- The performance of various machine learning models is analyzed on selected feature subsets to assess detection accuracy and computational efficiency.
- The robustness of the NIDS framework is validated using cross-validation sampling strategies and statistically using the Friedman test and Wilcoxon-Holm correction test.

1.4.2 Enhanced Anomaly-based NIDS Leveraging Modified Picture Fuzzy Clustering

For intrusion detection, a major challenge is the curse of high dimensionality, where an increasing number of features makes the data more sparse, making it difficult for models to identify meaningful patterns and relationships. Higher dimensionality often results in increased false alarm rates, greater computational complexity, and a higher risk of overfitting. Traditional clustering methods in NIDS face challenges with noisy data, overlapping attack patterns, and uncertain classifications. This study presents an efficient anomaly detection system that first addresses the class-imbalance issue using Synthetic Minority Over-sampling Technique (SMOTE), then reduces the dataset's dimensionality, and further classifies network traffic using supervised ML methods. Additionally, it clusters network attacks using an enhanced fuzzy clustering-based approach.

- We developed an anomaly-based NIDS using machine learning algorithms and a novel modified picture fuzzy clustering approach.
- Dimensionality reduction is applied to mitigate the impact of high dimensionality on the dataset and enhance overall performance by reducing time delay, computational cost, and resource consumption.

- Binary classification of network traffic using supervised ML classifiers on the dimensionality-reduced dataset, to distinguish the network attacks from the benign ones.
- We developed a novel picture fuzzy clustering-based technique to detect and cluster various network attacks present in the anomalous network.

1.4.3 Real-time NIDS based on Hybrid Incremental Learning

Real-time intrusion detection demands scalable solutions that adapt to evolving cyber threats, highlighting the need for adaptability. This research proposes a hybrid incremental learning framework for continuous and efficient NIDS operation.

- A NIDS framework is designed for real-time analysis and processing of captured network packets, ensuring efficient data streaming.
- Apache Kafka is leveraged for real-time data ingestion, facilitating scalable intrusion detection in dynamic network environments.
- We developed a Hybrid Incremental Learning-based IDS that combines a supervised incremental learning algorithm with an ensemble of unsupervised algorithms for adaptive and real-time network anomaly detection.
- Adaptive drift detection is performed by actively monitoring the network and retraining the model to sustain its performance in an evolving network environment.

In summary, this thesis presents innovative solutions to the multifaceted challenges of intrusion detection, including identifying various network attacks, selecting optimal features, addressing high dimensionality, and detecting intrusions in a dynamic network environment.

1.5 Organization of the Thesis

The content of this thesis is structured as follows:

- **Chapter 2** reviews existing approaches to anomaly-based intrusion detection, focusing on network traffic classification using supervised and unsupervised ML algorithms, Deep Learning algorithms, feature selection, dimensionality reduction, and real-time IDS. Furthermore, it discusses the research gaps and objectives.
- **Chapter 3** presents the proposed NIDS framework, which classifies network traffic data after selecting an optimized feature set using a novel feature selection technique.
- **Chapter 4** focuses on clustering various network attacks using the novel Picture Fuzzy Clustering-based approach, following the classification of dimensionality-reduced network traffic data.
- In **Chapter 5**, the development of a real-time NIDS leveraging hybrid incremental learning models for dynamic intrusion detection is discussed. The chapter explores how these models adapt to evolving threats, incorporate continuous learning, and enhance detection accuracy while minimizing false alarms.
- In **Chapter 6**, the key conclusions and achievements from each contribution are presented. Additionally, it discusses potential future research directions for intrusion detection in network environments.

Chapter 2

Literature Review

This chapter discusses well-established and state-of-the-art methodologies related to anomaly-based network intrusion detection systems. The literature review covers feature selection, dimensionality reduction, and real-time intrusion detection systems. Additionally, the chapter explores the limitations of existing frameworks, research objectives, and proposed solutions.

2.1 Related Work on Anomaly-based NIDS

Anomaly-based Network Intrusion Detection Systems (NIDS) play a crucial role in identifying malicious activities by analyzing deviations from normal network behavior. The concept of intrusion detection and threat surveillance was first proposed by J. P. Anderson in 1980 [5], wherein various computer security threats imposed on the system are discussed and how to monitor and detect such threats based on the anomalous behaviours present in the network.

This section presents a structured review of various machine learning-based techniques explored over the years, categorizing them into traditional ML methods with feature selection, dimensionality reduction approaches, fuzzy clustering-based techniques, and real-time IDS solutions.

2.1.1 Machine Learning Methods for Intrusion Detection Systems Using Feature Selection

Feature selection (FS) is a fundamental yet important step in improving the efficiency and accuracy of ML-based IDS. [29] compares the performance of Random Forest, SVM, and Extreme Learning Machine (ELM) in detail for intrusion detection, analyzing their performance against the NSL-KDD data set to determine their accuracy in detecting intrusions. It was observed that ELM performed better than the rest when full samples were under consideration. In [30], the authors developed a Semi-supervised Multi-Layered Clustering (SMLC) model for preventing and detecting network intrusions. SMLC is capable of learning from partially labelled data while proving to be superior to the tri-training and supervised ensemble Machine Learning models such as Bagging, AdaBoostM1, etc., on the network intrusion datasets, NSL and Kyoto 2006+.

Tao *et al.* [31] proposed an intrusion detection algorithm, FWP-SVM-GA, based on the Genetic algorithm (GA) and Support Vector Machine (SVM) algorithm, where the algorithm initially does feature selection, weight and parameter optimization of SVM based on GA. The GA-based feature selection method is implemented to reduce the SVM error rate and enhance the true positive rate. A performance analysis of IDS with the help of a feature reduction technique was performed by Kasongo *et. al.* [32] on the UNSW-NB15 dataset. The authors state that a decline in performance was observed as the dimensionality of the data space increased. Their experiment implemented five ML techniques, SVM, kNN, LR, ANN, and DT, on the reduced feature set and determined that the XGBoost-DT combination obtained the most efficient results. [33] presents a comparative study of six ML algorithms, namely, Naïve Bayes, Neural Network, k-NN, SVM, and Decision Tree for intrusion detection in association with four feature selection techniques, CFS, Information Gain Ratio (IGR), PCA, and Minimum Redundancy Maximum Relevance feature selection (mRMR). It was concluded that it was difficult to choose one ML technique over the other, and k-NN performed better than the other ML methods, and IGR performed better than the rest.

Kocher and Ahuja [34] presented a performance analysis of ML classifiers on the UNSW-NB15 dataset. The popular filter-based Chi-squared feature selection method was employed to reduce redundant features. It was observed that the random forest classifier outperformed the others over the original as well as the selected optimal feature

set. Uzun and Ball [35] introduced a novel method for enhancing intrusion detection systems by combining multivariate outlier detection with ReliefF feature selection on the NSL-KDD dataset. With the Random Forest algorithm achieving the highest accuracy of 99.21%, the study concludes that combining ReliefF feature selection with outlier detection significantly improves intrusion detection accuracy and reduces processing time. In [36], the authors evaluate various machine learning classifiers using the NSL-KDD dataset for intrusion detection, including the k-nearest neighbour, decision tree, naïve Bayes, logistic regression, random forest, and ensemble methods. They apply a basic rule-based feature selection approach to enhance efficiency by reducing dataset size and computational complexity. Results show that the ensemble approach achieves the highest accuracy of 99.5%, concluding that the ensemble methods, with feature selection, are a promising approach for effective intrusion detection systems. An efficient IDS was proposed by [37], where initially, data sampling and feature selection are performed on the UNSW-NB15 dataset using iForest and a genetic algorithm, respectively. Subsequently, the random forest classifier-based IDS is developed for intrusion detection. A comprehensive performance analysis on a combination of three feature selection techniques and seven ML models on the NSL-KDD dataset is discussed in [38]. The results concluded that feature selection can significantly improve the accuracy and speed of attack classification.

[39] introduced an enhanced Genetic Algorithm-based feature selection method, GbFS, aimed at preserving unique data information with minimal features to improve classifier accuracy for network security and intrusion detection. Tested on three benchmark datasets, CIRA-CIC-DOHBrw-2020, UNSW-NB15, and Bot-IoT, GbFS shows a significant improvement in the accuracy of the ML classifiers, achieving up to 99.80% detection rate. Kaushik *et al.* [40] compares the performance of the ML classifiers with the ensemble approaches for intrusion detection in the military network dataset. Experimental results suggest that ensemble models provide better detection accuracy when feature selection is employed. The authors in [41] propose an effective NIDS framework, using recursive feature elimination with cross-validation (DT-RFECV), to select the optimal set of features from the UNSW-NB15 data set. The performance was evaluated using various ML classifiers, and it was concluded that the model achieved higher accuracy when the entire feature set was used.

Another network-based IDS was presented by [42], where, after pre-processing the

datasets, CSE-CIC-IDS2018 and UNSW-NB15, extracting the features, a feature selection technique, Opposition-based Northern Goshawk Optimization algorithm (ONgO), is used to find the optimal features. Various attacks were detected with the hybrid M-multiSVM. Compared with other ML models, the proposed hybrid model achieved higher accuracy than the rest. An ensemble feature selection approach, IDS-EFS, is proposed by [43]. The experimental results on the KDDCup'99 suggest that, compared to the other feature selection methods, the proposed method gave better results. In [44], an ensemble model approach was used in the CSE-CIC-IDS2018 dataset to compare the performance of seven individual classifiers. The objective was to identify the top-performing models and integrate them into a classifier unit for evaluation. Naïve Bayes (NB), RF, Decision Trees (DT), Quadratic Discriminant Analysis (QDA), Logistic Regression (LR), Multilayer Perceptron (MLP), and Gradient Boosting were the seven classifiers. The ensemble model was observed to perform better than the individual classifiers with a precision and accuracy of 0.988 each when 23 feature subsets were used, and the accuracy and precision of 0.987 and 0.980, respectively, were achieved in the original data set.

Mallampati and Hari [45] presented another feature selection method based on feature importance, employing a fusion of various statistical feature importance techniques, including filter, wrapper, and embedded feature selection approaches. The experiments were conducted on the NSL-KDD, UNSW-NB15, and CIC-IDS2017 datasets. To address the class imbalance issue, they utilized the Adaptive Synthetic oversampling technique. The results indicated that classifier performance was better on balanced datasets than on imbalanced ones. The experimental findings demonstrated that the XGBM model outperformed all others across the three datasets. In [46], the authors utilized Gini feature importance and Recursive Feature Elimination (RFE) for feature selection on the NSL-KDD dataset, while applying feature importance-based selection on the CIC-IDS2017 dataset. Testing various ML models on the KDD Train+ test dataset revealed that models using feature importance-based selection and 10-fold cross-validation exhibited improved performance.

Based on the above discussion, feature selection consistently emerges as a critical step for improving the performance of ML-based intrusion detection systems. Across studies, its role in reducing dimensionality, removing irrelevant attributes, and enhancing both classification accuracy and processing speed is widely acknowledged [29, 31,

34, 38, 45].Notably, models with optimized feature sets often outperform those using the complete feature set, especially in high-dimensional datasets like CIC-IDS2017. This underscores the importance of effective feature curation in developing IDS frameworks that can scale effectively. Studies using older datasets like NSL-KDD often report very high accuracies, sometimes exceeding 99%, likely due to the dataset's reduced complexity. However, newer and more challenging datasets like UNSW-NB15 and CSE-CIC-IDS2018 offer a more realistic evaluation environment [29, 36, 44].

The performance and usability of filter, wrapper, and embedded feature selection techniques vary significantly, making it essential to consider their practical applicability. Filter-based methods such as Chi-square and ReliefF offer rapid preprocessing and are well-suited for real-time scenarios, though they may overlook feature interactions [34, 35]. In contrast, wrapper-based methods like GA and RFE yield more accurate models by evaluating feature subsets with classifiers, but are computationally intensive [31, 41, 46]. Whereas embedded approaches maintain a balance, making them a promising choice for operational IDS environments.

Table 2.1 summarizes research on intrusion detection using various filter or wrapper-based feature selection techniques, as well as approaches without feature selection. It also lists the datasets used, the classification models employed, and the best results obtained in each study.

2.1.2 Machine Learning Approaches for Network Attack Detection

In this subsection, we review existing ML-based IDS that focus on detecting various network attacks. We categorize these approaches based on their use of with or without dimensionality reduction (DR) methods and clustering-based anomaly detection. Dimensionality reduction techniques enhance performance by capturing essential network characteristics. Additionally, clustering-based techniques, such as fuzzy clustering, aid in distinguishing normal and malicious traffic without requiring labeled data.

For intrusion detection, a model generalization [47] was achieved by using four unsupervised ML techniques, namely, autoencoder, one-class SVM, isolation forest, and PCA on CIC-IDS-2017 and CSE-CIC-IDS-2018. The strategy used in this study enables the evaluation of the model's ability to perform well on diverse datasets beyond the ones they were initially trained on. In terms of accuracy and precision, the autoencoder

Table 2.1: Summary of existing ML-based models for network intrusion detection

Ref.	Dataset	Detection Technique	FS Approach	FS Technique	Performance/Results
[29]	NSL-KDD	SVM, Random forest, ELM	-	-	ELM outperforms Acc: 99.5%, Prec: 98.6%
[31]	KDD Cup'99	SVM	Wrapper	Genetic Algorithm	On 9 selected features DR: 96.61%, False Positive Rate: 3.39
[30]	NSL, Kyoto 2006+	SMLC model based on K-Means algorithm	-	-	NSL: Acc: 99.58% Kyoto 2006+: Acc: 99.39%
[33]	NSL-KDD	Naive Bayes, SVM, Decision Tree, Neural Network, k-NN	Filter	CFS, Information Gain Ratio (IGR), PCA, Mrmr	IGR with k-NN Acc: 99.07%
[37]	UNSW-NB15	Proposed DO_IDS	Wrapper	Genetic Algorithm	Acc: 0.928 FAR: 0.033
[32]	UNSW-NB15	SVM, kNN, LR, ANN, DT	Filter	XGBoost	XGBoost with Decision tree (19 features selected) Acc: 90.85% for binary classification scheme
[34]	UNSW-NB15	LR, NB, RF, SGD, KNN	Filter	Chi-squared	RF with Chi-squared: Acc: 99.64%
[36]	NSL-KDD	kNN, DT, NB, LR, RF	Filter	Proposed rule-based feature selection	Acc: 99.5%
[38]	NSL-KDD	SVM, NB, DT, RF, k-NN, LR, and Artificial Neural Networks	Filter	Chi-Square, Information Gain, and Recursive Feature Elimination (RFE)	RFE outperformed the other FS methods and SVM gave better results for all attack categories
[39]	CIRA-CIC-DOHBrw-2020, UNSW-NB15, and Bot-IoT	SVM, k-NN, XgBoost	Wrapper	Proposed GbFS	In all three datasets: Highest ACC: 99.80%
[35]	NSL-KDD	Bagging, Bayes Network, Filtered Classifier, kNN, C4.5, RF	Filter	ReliefF	RF with ReliefF (20 selected features): Acc: 99.2187%
[40]	Military network dataset	LR, k-NN, NB, DT, SVM, RF, Gradient Boost (GB), Extreme GB, AdaBoost	Filter	Chi-squared and Information Gain (IG)	RF + Chi-squared: Acc: 96.65% RF + IG: Acc: 99.12%
[41]	UNSW-NB15	LR, NB, RF, AdaBoost, Stochastic Gradient Descent, Multi-layer Perceptron	Filter	recursive feature elimination with cross-validation using a decision tree model as an estimator (DT-RFECV)	RF + DT-RFECV: Acc: 95.30%
[42]	CSE-CIC-IDS2018, UNSW-NB15	Mud Ring assisted multilayer SVM (M-MultiSVM)	Wrapper	Opposition-based Northern Goshawk Optimization algorithm (ONgO)	CSE-CIC-IDS2018: Acc: 99.89% UNSW-NB15: Acc: 97.535%
[43]	KDDCup'99	LR, RF, KNN, DT, SVM	Filter	Proposed IDS-EFS	Acc: 99%
[44]	CSE-CIC-IDS2018	Ensemble learning model	Filter	Chi-square and Spearman's rank correlation coefficient	Selected features (23): Acc: 0.988, Prec: 0.988
[45]	NSL-KDD, UNSW-NB15, CIC-IDS2017	Extra Tree, SVM, LR, DT, XGBM	Feature Importance	Proposed fusion FS	NSL-KDD (9 features): Acc: 99.86% CIC-IDS2017 (7 features): Acc: 99.68% UNSW-NB15 (8 features): Acc: 92.4%
[46]	NSL-KDD, CIC-IDS2017	DT, RF, NB, KNN, SVM, XGBoost, Ensemble	Feature Importance	RFE and Gini Importance	NSL-KDD: SVM outperform others CIC-IDS2017: DT and Ensemble model outperforms

achieved the highest results of 0.9426 and 0.9459, respectively, on the CIC-IDS-2017 dataset, whereas one-class SVM on CSE-CIC-IDS2018 achieved an accuracy of 0.8898 and a precision of 0.9268, which outperformed the rest. In [48], the authors propose an ensemble model of SVM, MLP, and an instance-based learning method (IBK) to classify network attacks and normal traffic flows in three datasets, NSL-KDD, Kyoto 2006+, and ISCX 2012. Information gain (IG), along with PCA, was used to reduce the dimension space of each of the datasets. The experimental results show that the proposed model in the mentioned datasets achieved the highest accuracy of 98.24%, 98.95%, and 99.01%, respectively. In [49], on the CSE-CIC-IDS2018 dataset, six ML techniques were implemented, namely, Decision Tree, AdaBoost, Random Forest, K-Nearest Neighbour (K-NN), LDA, and Gradient Boosting. The experimental analysis demonstrated that the implemented models achieved a significantly high level of accuracy, outperforming recent literature benchmarks. The authors also classified various network attacks like Botnet attacks, Denial of Service (DoS), Infiltration, SQL injection, and Brute Force attacks. Out of the six ML models, AdaBoost attained the highest accuracy of 99.69%.

Performance evaluation of three ML classifiers, RF, MLP, and Long-Short Term Memory (LSTM), that follow a sequential approach, is proposed by [50] on the CIDDS-001 dataset. On the basis of experimental analysis, it is suggested that addressing anomaly detection from a sequential perspective may yield better results. The proposed study also classifies the network traffic dataset into various types of attacks, such as brute force attacks, DoS, ping scans, and port scan attacks. The LSTM model proves to be highly reliable in capturing sequential patterns within network traffic data, exhibiting an impressive accuracy of 99.94%. [51] proposed a dynamic auto-selection classifier adopted on various ML techniques to enhance the model's performance by using every technique's capabilities for better detection of attacks. When considering a balanced dataset, certain attacks, such as worms, generic attacks, and DoS attacks, are observed to demonstrate a true positive rate (TPR) surpassing 90%. Compared to individual ML models, the proposed model attained the highest accuracy of 87.6%. In [52], an IDS implemented with ML techniques along with dimensionality reduction was presented. The CICIDS2017 dataset was used and, initially, two-dimensionality reduction techniques, Autoencoder and PCA, were employed. Various ML classifiers such as RF, Bayesian Network, QDA, and LDA were used to classify intrusions in the network data. Using dimensionality reduction methods, the feature set was brought down from 81 to 10

while depicting the highest accuracy of 99.6%. The proposed model gave the following results: RF with Autoencoder provided an F-measure of 0.995, while RF with PCA on the unbalanced dataset had an F-measure of 0.996, and RF with PCA on the balanced dataset resulted in an F-measure of 0.988. In [53], six ML models, namely, DT, RF, SVM, Artificial Neural Network (ANN), Deep Neural Network (DNN), and Naïve Bayes on three different IDS datasets, CIC-IDS2017, CSE-CIC-IDS2018, and LUFlow dataset. For feature or dimensionality reduction, the authors used the RandomForestClassifier. The overall performance of the ML classifiers on CIC-IDS2017 displayed a higher accuracy of 0.9967 using RF, and on the CSE-CIC-IDS2018, SVM had the highest accuracy of 0.7559, and in the LUFlow dataset, RF and DT both had the highest detection accuracy of 0.9994.

[54] proposed a novel NIDS based on the Difficult Set Sampling Technique (DSSTE) algorithm, where various classifiers such as RF, SVM, XGBoost, AlexNet, LSTM, and Mini-VGGNet were used to detect various threats present in the CSE-CIC-IDS2018 network dataset. The DSSTE algorithm was performed to overcome the problem of class imbalance. The performance of 24 models was compared, and it was concluded that the proposed DSSTE algorithm, along with MiniVGGNet, achieved the highest accuracy of 96.99%. [55] designed a Class-Wise Focal Loss (CWFL) and Variational AutoEncoder (VAE)-based class balancing approach integrated with XGBoost. It also presents a comparative analysis of the proposed method with the existing class balancing approaches on the NSL-KDD and CSE-CIC-IDS2018 datasets, demonstrating that the proposed framework has a higher precision of 99.67% than the traditional ones.

[56] proposed a Fuzzy C-Means-based intrusion detection model called Robust Spatial Kernel Fuzzy C-Means (RSKFCM). To hand-pick the most discriminated features, the authors implemented PCA, and then to cluster the network data, RSKFCM was employed. The model's performance was compared with versions of conventional Fuzzy C-Means (FCM) techniques and drew an inference that the proposed model outperformed the rest with an accuracy of 86.26% and a False Positive Rate (FPR) of 17.04. [57] proposes a hybrid approach of clustering and classification using the Gaussian Mixture Models (GMM) and K-Means clustering technique with the random forest classifier, where detection is done on two benchmark datasets, NSL-KDD and KDD Cupp99. The proposed hybrid model, with the help of RF, is able to classify various types of attacks such as DoS, probe attack, U2R, and R2L attacks. The hybrid K-Means and

RF model on NSL-KDD achieved an accuracy of 99.85%, and GMM and RF on KDD Cup99 were implemented, achieving an accuracy of 98.27%. On the other hand, a supervised ensemble learning model comprising K-NN and SVM models presented a higher performance in detecting individual network attack types than the individual classifier and clustering models implemented.

[58] compared two clustering techniques, that is, K-Means and Fuzzy C-Means clustering to detect network attacks using the NSL-KDD dataset. The algorithms were able to detect major attacks like probe attacks, DoS, U2R, and R2L attacks. The experimental results show that Fuzzy C-Means detected 45.95% of the attacks, and K-Means detected 44.72% of attacks. A comparative study was done [59], where a supervised and an unsupervised ML technique comparison was performed on the KDD Cup99 dataset along with the Chi-squared feature selection method. The supervised ML technique Support Vector Machine (SVM) and the unsupervised ML technique Fuzzy Kernel C-Mean (FKCM) was implemented for this purpose. Experimental results were carried out on only 10 features out of 41. In terms of the highest accuracy achieved, FKCM secured 80.29% accuracy along with radial basis function parameter 0.5, whereas SVM and polynomial kernel attained 88.88% accuracy with parameter 3.

A hybrid attack detection model was proposed by [60] on the NSL-KDD dataset. The model works in 2 stages; first, in the anomaly detection phase, FCM is used to cluster the normal data and attacks. After the attacks are detected, in the misuse detection phase, the known attacks are classified with the help of Classification and Regression Trees (CART) and the isolation forest to detect the unknown attacks. The experimental results show that the classification of normal data and attacks by the proposed model achieved an accuracy of 0.8454. Meanwhile, the unknown attack detection accuracy is as follows: DoS: 0.8137, Probe: 0.8793, U2R: 0.8288, and R2L: 0.7176.

It is observed across the studies that dimensionality reduction techniques such as PCA and autoencoders have proven effective in enhancing the performance of ML classifiers by reducing computational complexity and highlighting relevant patterns in high-dimensional network traffic data [48, 52, 53]. Clustering-based methods, particularly fuzzy clustering, such as RSKFCM and FKCM, have been explored for their ability to identify novel attacks without labelled data and to capture the uncertainty and overlapping nature of network traffic patterns, yet they often fall short in handling ambiguity in highly imbalanced or complex attack scenarios [56, 58, 59, 60]. While many models

report very high accuracy on benchmark datasets, their performance often varies significantly across datasets, for instance, CIC-IDS2017 and LUFlow, revealing potential overfitting or lack of generalizability [53, 55].

Table 2.2 summarizes the related work reviewed in this subsection, presenting existing IDS models for detecting various network attacks, using dimensionality reduction. This summary also presents studies conducted using the traditional clustering and fuzzy clustering approaches for intrusion detection

2.1.3 Deep Learning Approaches for Intrusion Detection

In recent years, deep learning (DL) techniques have gained significant attention in the field of intrusion detection due to their ability to automatically learn complex patterns from raw or high-dimensional data. Models like convolutional neural networks (CNNs), recurrent neural networks (RNNs), and autoencoders have shown promising performance, particularly in detecting sophisticated or previously unseen attacks. The following subsection reviews recent DL-based and hybrid intrusion detection approaches.

In [61], a DL-based IDS using various RNN architectures such as simple RNN, LSTM, and Gated Recurrent Unit (GRU) was proposed. The framework was further enhanced by performing feature selection using XGBoost, which improved network security against evolving threats. Tested on NSL-KDD and UNSW-NB15 datasets, the model selected 22 and 17 key features, respectively. For binary classification, XGBoost-LSTM achieved 99.49% accuracy on NSL-KDD, while XGBoost-RNN performed best on UNSW-NB15 with 87.07%. The experimental results indicated that combining the XGBoost feature selection method with RNNs optimised the performance of the framework for both binary and multiclass intrusion detection. In multiclass classification, XGBoost-LSTM excelled on NSL-KDD and XGBoost-GRU on UNSW-NB15, demonstrating strong performance across both datasets.

[62] developed a hybrid DL approach combining CNN and bidirectional LSTM (BiLSTM) to perform both binary and multiclass network intrusion detection. The performance of the framework was evaluated in a software-defined network (SDN) environment and tested on three datasets, NSL-KDD, UNSW-NB15, and InSDN. To further enhance the effectiveness of the proposed model, a hybrid feature selection technique based on Random Forest and Recursive Feature Elimination (RFE) was employed.

Table 2.2: Summary of existing models for network attack detection

Ref.	Dataset	Technique	DR Approach	Attacks Detected	Performance/Results
[56]	KDD Cup99	Robust Spatial Kernel Fuzzy C-Means	PCA	-	Acc: 86.26%, False Positive Rate: 17.04
[54]	NSL-KDD, CSE-CIC-IDS2018	RF, SVM, XGBoost, AlexNet, LSTM, Mini-VGGNet	-	Bot, DoS, DDoS, Brute Force, Infiltration, SQL Injection	NSL-KDD: (DSSTE-AlexNet) Avg. Acc: 82.84% CSE-CIC-IDS2018: (DSSTE-miniVGGNet) Avg. Acc: 96.99%
[47]	CIC-IDS-2017, CSE-CIC-IDS2018	Autoencoder, one-class SVM, Isolation forest, PCA	-	-	CIC-IDS-2017: (Autoencoder) Acc: 0.9426, Prec: 0.9459 CSE-CIC-IDS2018: (One-class SVM) Acc: 0.8898, Prec: 0.9268
[48]	NSL KDD, Kyoto 2006+, ISCX 2012	Ensemble: SVM, MLP, IBK	Info. Gain + PCA	-	NSL-KDD: Acc: 98.24%, FAR: 0.017 Kyoto 2006+: Acc: 98.95%, FAR: 0.021 ISCX 2012: Acc: 99.011%, FAR: 0.01
[49]	CSE-CIC-IDS2018	ADA, DT, RF, K-NN, GB, LDA	-	Bot, DoS, Brute Force, Infiltration, SQL Injection	Multi-classification Avg. Acc: GB: 99.38%
[55]	NSL-KDD, CSE-CIC-IDS2018	XIDINTFL-VAE	-	DDoS, DoS, Brute Force, Botnet, Infiltration, Web attack	NSL-KDD: Acc: 99.79% CSE-CIC-IDS2018: Acc: 99.89%
[50]	CIDDS-001	MLP, RF, LSTM	-	DoS, Brute Force, Ping Scan, Port Scan	LSTM Acc: 99.94%, F1: 91.66%
[51]	UNSW-NB15	K-NN, DT, RF, SVM, XGBoost, MLPNN, LSTMNN	-	Analysis, Backdoor, DoS, Exploits, Fuzzer, Generic, Reconnaissance, Shellcode, Worms	Dynamic Classifier on balanced dataset: Acc: 87.6%
[57]	KDD Cup99, NSL-KDD	RF, K-NN, Gaussian Mixture Model	-	DoS, Probe, R2L, U2R	NSL-KDD: Acc: 99.85% KDD Cup99: Acc: 98.27%
[52]	CIC-IDS2017	RF, Bayesian Network, LDA, QDA, Autoencoder, PCA	Autoencoder, PCA	DoS, DDoS, HeartBleed, PortScan, Brute Force, Web attacks, Infiltration, Botnet	RF + Autoencoder: F-measure: 0.995 RF + PCA(Original dataset): F-measure: 0.996
[53]	CIC-IDS2017. CSE-CIC-IDS2018, LUFlow	DT, RF, SVM, Naïve Bayes, ANN, DNN	RandomForestClassifier	-	CIC-IDS2017: RF-Acc: 0.9967 CSE-CIC-IDS2018: SVM-Acc: 0.7559% LUFlow: DT & RF Acc: 0.9994
[58]	NSL-KDD	K-Means, Fuzzy C-Means clustering	-	DoS, Probe, R2L, U2R	K-Means: 44.72% of attacks detected Fuzzy C-Means: 45.95% of attacks detected
[59]	KDD Cup99	SVM, Fuzzy Kernel C-Means (FKCM)	-	DoS, Probe, R2L, U2R	Highest accuracies: SVM + polynomial kernel (parameter 3) Acc: 88.88% FKCM + radial basis function (parameter 0.5) Acc: 80.29%
[60]	NSL-KDD	FCM, CART, Isolation forest	-	DoS, Probe, R2L, U2R	Binary Classification: Acc: 0.8454 Unknown attack detection accuracy: DoS: 0.8137 Probe: 0.8793 U2R: 0.8288 R2L: 0.7176

The experimental outcomes suggested that the proposed CNN-BiLSTM model outperformed the others by achieving the highest binary classification accuracy of 97.77% in the InSDN dataset, 95.96% in NSL-KDD, and 93.51% in the UNSW-NB15 dataset. For multiclass classification, it attained the best accuracy with 97.12%, 98.42%, and 84.23% for InSDN, NSL-KDD, and UNSW-NB15 datasets, respectively.

With growing volume and complexity of network traffic, Sajid et. al. [63] proposed a hybrid intrusion detection model that combines traditional machine learning with deep learning techniques. Their approach integrates XGBoost and CNN for efficient feature extraction, followed by LSTM networks for classification. The model was analysed on four datasets, NSL-KDD, UNSW-NB15, CIC-IDS2017, and WSN-DS, for both binary and multiclass classification tasks. The results showed promising performance, with the hybrid model achieving high detection rates and maintaining a low FAR. The CNN-LSTM combination achieved 96.21% accuracy on the CIC-IDS2017 dataset for binary classification, while the XGBoost-LSTM model reached 94.41% test accuracy on the NSL-KDD dataset. The study highlights the model's ability to generalize well across datasets, effectively detect various types of attacks, and minimise false positives, primarily due to its strong feature selection process and the use of sequential learning for improved pattern recognition.

Wang et. al. [64] proposed and evaluated a DL-based intrusion detection framework using the up-to-date CSE-CIC-IDS2018 dataset. The authors compared the performance of six DL models, DNN, CNN, RNN, LSTM, CNN+RNN, and CNN+LSTM, for both binary and multiclass classification of network traffic. Comprehensive data preprocessing was performed to handle redundancy, outliers, and inconsistent formats. Results demonstrated that all models achieved over 98% accuracy, with individual models, DNN, CNN, and RNN, offering faster inference times compared to the hybrid ones. Among the six models, CNN+LSTM showed the best detection for minority classes like infiltration, while simpler models were better suited for real-time implementation due to lower computational overhead. Furthermore, CNN+RNN and CNN+LSTM models achieved the highest accuracy of 98. 84% for multiclass classification.

Another DL model based on attention for intrusion detection was proposed by [65] that leverages the strength of CNN, BiLSTM, and a multi-head attention mechanism. This architecture was designed to effectively capture both spatial and temporal features, while the attention layer emphasises the most relevant parts of the input for im-

proved classification. The model was evaluated on NSL-KDD, UNSW-NB15, and CICIDS2017 datasets. The proposed model achieved impressive performance, with the highest accuracies of 99.87%, 99.64%, and 99.72% on NSL-KDD, UNSW-NB15, and CIC-IDS2017 datasets for binary classification, outperforming traditional CNN, LSTM, and hybrid models.

[66] proposed a hierarchical DL model based on LSTM and attention mechanisms, which was analysed on the UNSW-NB15 dataset using various traditional ML and DL-based approaches. The experimental outcomes show that the proposed model was less time-consuming and achieved a detection rate of 92.2% for binary classification. A deep neural network (DNN) IDS was proposed by [67] to enhance network security by addressing the challenges of classifying diverse attack types in the NSL-KDD dataset. The model achieved a training accuracy of 91.30% and a validation accuracy of 94.38%, with low loss values, demonstrating effective learning and generalization.

A comparative study of various DL models, CNN, DNN, RNN, LSTM, GRU, and hybrid CNN-LSTM was conducted by Elsayed et. al. on the benchmark NSL-KDD dataset [68]. Their experimental findings reveal that GRU outperformed the others by achieving a detection accuracy of 99.54% for binary classification, while LSTM gave the best detection results of 99.39% for multiclass classification. In another study, the authors [69] propose a multistage AI-enabled intrusion detection framework combining a DNN classifier and two autoencoders to detect known, zero-day, and adversarial attacks. The framework integrates three DNN components, one classifier and two specialized autoencoders. Using transfer learning with one-shot learning, the model freezes selected layers to enhance adaptability. Validated on benchmark datasets, the framework achieved an average accuracy of 98.5%, demonstrating strong performance against evolving threats. [70] introduces a study ENIDS, a DL-based ensemble model for detecting various types of cyberattacks. It combines three base models, namely, CNN, LSTM, and GRU, under a DNN-based meta learner. Evaluated on the UNSW-NB15 and CIC-IDS2017 datasets, ENIDS achieved 90.6% on UNSW-NB15 and 99.6% accuracy on the CIC-IDS2017 dataset. Experimental results show that ENIDS outperforms existing deep learning models in both detection performance and computational efficiency.

An intelligent NIDS based on and optimized CNN-LSTM model was proposed by [71]. SMOTE was employed to deal with the data imbalance issue. The hybrid model

trained on the UNSW-NB15 dataset, outperformed the others by achieving the highest detection accuracy of 78.47% for binary and 78.36% for multiclass classification. Aljehane et. al.[72] proposed a novel deep learning model, GJOADL-IDSNS, for intrusion detection. The model is based on an attention-enabled BiLSTM architecture and is integrated with the Golden Jackal Optimization algorithm for feature selection. On the CIC-IDS2017 dataset, the proposed model attained its highest accuracy of 99.70%.

The reviewed studies highlight the growing effectiveness of deep learning in intrusion detection, particularly through hybrid models combining CNN, RNN, and attention mechanisms. Integrating DL with feature selection techniques like XGBoost and Random Forest consistently improved accuracy across datasets such as NSL-KDD, UNSW-NB15, and CIC-IDS2017. Despite their strengths, DL models come with notable challenges. They often require large amounts of labelled data, substantial computational resources, and remain difficult to interpret. Their longer training times and sensitivity to hyperparameter tuning can also hinder timely deployment, especially in real-time monitoring systems where adaptability and low latency are critical. These limitations make them less practical for real-time or resource-constrained environments. Overall, DL methods show strong potential for improving intrusion detection systems, especially when paired with efficient preprocessing and robust feature selection strategies.

Table 2.3 provides a comparative overview of the performance of various deep learning-based studies discussed in this subsection for performing binary and multiclass classification of network traffic data for intrusion detection.

2.1.4 Real-time Intrusion Detection System

With the increasing complexity and volume of network traffic, real-time intrusion detection has become a critical area of research. Traditional ML-based IDS often struggle with high computational costs and delayed detection, making them unsuitable for real-time applications. This subsection discusses the existing methodologies and frameworks designed to enhance the responsiveness and scalability of IDS in real-world environments.

[73] presents a real-time NIDS based on a deep learning model, AE-AlexNet. They use Flume to collect the logs and Flink to clean the logs in real time. The experiments suggested that with more training, the model's performance can be enhanced. This pro-

Table 2.3: Summary of existing DL-based models for network intrusion detection

Ref.	Dataset	Detection Technique	FS Technique	Classification Task	Performance/Results
[61]	NSL-KDD, UNSW-NB15	Simple RNN, LSTM, GRU	XGBoost	Binary, Multiclass	NSL-KDD: Binary: XGBoost-LSTM: 99.49%, Multiclass: XGBoost-LSTM: 86.93% UNSW-NB15: Binary: XGBoost-RNN: 87.07%, Multiclass: XGBoost-LSTM: 78.40%
[62]	NSL-KDD, InSDN, UNSW-NB15	CNN-BiLSTM	Random Forest+RFE	Binary, Multiclass	InSDN: Binary Acc: 97.77%, Multiclass Acc: 97.12% NSL-KDD: Binary Acc: 93.51%, Multiclass: 98.42% UNSW-NB15: Binary Acc: 93.51%, Multiclass: 84.23%
[63]	NSL-KDD, WSN-DS, UNSW-NB15, CIC-IDS2017	LSTM	CNN, XGBoost	Binary, Multiclass	NSL-KDD and UNSW-NB15 Highest Acc: 98.40%
[64]	CSE-CIC-IDS2018	DNN, CNN, RNN, LSTM, CNN+RNN, CNN+LSTM	-	Binary, Multiclass	Binary (Highest Acc.): CNN+LSTM: 98.85% Multiclass (Highest Acc): CNN+LSTM, CNN+RNN: 98.84%
[65]	NSL-KDD, UNSW-NB15, CIC-IDS2017	CNN, BiLSTM, attention mechanism	Extra tree classifier	Binary	Highest Accuracy: NSL-KDD: 99.87%, UNSW-NB15: 99.64%, CIC-IDS2017:99.72%
[66]	UNSW-NB15	LSTM, attention mechanism	-	Binary	Highest Accuracy: 92.20%
[67]	NSL-KDD	DNN	-	Binary	Highest Training Accuracy: 91.30% Highest Validation Accuracy: 94.38%
[68]	NSL-KDD	CNN, DNN, RNN, LSTM, GRU, hybrid CNN-LSTM	-	Binary, Multiclass	Binary (Highest Accuracy): GRU: 99.54 Multiclass (Highest Accuracy): LSTM: 99.39%
[69]	CIC-IDS2017, CSE-CIC-IDS2018	DNN	-	Multiclass	Highest Accuracy: 98.5%
[70]	UNSW-NB15, CIC-IDS2017	Ensemble: CNN, DNN, LSTM, GRU	-	Binary	Highest Accuracy: UNSW-NB15: 90.06%, CIC-IDS2017: 99.6%
[71]	UNSW-NB15	Optimized CNN-LSTM	-	Binary, Multiclass	Binary (Highest Accuracy): 78.47% Multiclass (Highest Accuracy): 78.36%
[72]	CIC-IDS2017	Proposed: GJOADL-IDSNS	Golden Jackal Optimization Algorithm	Multiclass	Highest Accuracy: 99.70%

posed model achieved the highest accuracy of 94.32%. An incremental learning model based on Naïve Bayes and SVM was proposed in [74], where the authors performed the experiments on the UAV intrusion detection dataset, which consists of six different real-time datasets. Amongst the two, Naïve Bayes achieved the highest accuracy of 0.9840. Another incremental learning-based NIDS was presented by [75], where the IDS was trained on the intrusion dataset UNSW-NB15. A comparative analysis of the incremental model and random forest in this study shows that the incremental learning model had 32% less training time than the random forest classifier.

Mirsky *et al.* [76] introduced Kitsune, a neural network-based NIDS designed for real-time network intrusion detection, distinguishing between normal and anomalous network traffic. This unsupervised learning model continuously monitors network activity and identifies anomalies using an ensemble of autoencoders. The model was evaluated across multiple attack scenarios in an IP camera surveillance network and an IoT environment. It achieved high detection accuracy with a low false positive rate, demonstrating a notable improvement over traditional feature-engineering-based approaches. In [77], a distributed intrusion detection framework based on the random forest algorithm is proposed for real-time analysis of network traffic captured using NetFlow. The framework integrates the random forest classifier with the Apache Spark distributed processing system to enable efficient real-time detection. The model's performance was evaluated against existing intrusion detection systems using the CIC-IDS-2017 dataset, where various machine learning techniques were compared. The results indicated that the gradient boosting decision tree achieved the highest detection performance. In [78], a performance analysis of an autoencoder-based intrusion detection system is conducted in a real-time environment using Apache Kafka and Spark Streaming. Kafka is utilized for real-time ingestion of network traffic transactions, while Spark Streaming processes the data in batches. The study observes that increasing the number of partitions leads to a slight reduction in processing time across three different batch configurations. The results indicate that a 50-second batch interval outperforms a 10-second batch interval in terms of efficiency.

A clear trend across studies is the integration of real-time data processing frameworks such as Apache Kafka, Apache Flink, and Spark Streaming to handle the high throughput and low-latency requirements of modern network environments [73, 77, 78]. These tools enable not just rapid ingestion and preprocessing, but also support batch and

micro-batch operations that align well with model inference times. Multiple approaches explore incremental learning strategies to maintain IDS adaptability in evolving traffic scenarios, highlighting their importance in balancing performance and speed [74, 75]. In contrast, batch-trained models [73] and ensemble autoencoders [76] offer strong initial detection but lack built-in adaptability unless retraining is performed. This highlights a clear trade-off between adaptability and performance that the incremental models are well-suited for dynamic environments, whereas batch models typically excel in more static scenarios.

2.2 Research Gaps

From the discussion in Section 2.1, numerous intrusion detection systems have been proposed that perform feature selection for enhancing the performance of the model, but to the best of our knowledge, there are limited studies that perform intrusion detection on the novel and current network attacks, using the combination of supervised and unsupervised machine learning algorithms that leverages the strengths of both methods, resulting in a more robust and adaptable system. Many IDS models rely on arbitrary feature selection techniques without standardization, leading to inconsistent results across different datasets. A systematic feature selection framework can help improve reproducibility and benchmarking.

Furthermore, as far as our knowledge extends, the majority of existing studies concentrate on utilizing conventional ML techniques to construct IDSs to detect various types of network attacks. On the other hand, some existing research considers an ensemble approach for enhancing the efficiency of intrusion detection. However, these conventional ML-based frameworks are typically effective only on small and low-dimensional datasets, often struggling to handle high-dimensional and large-scale data effectively. It is also observed that there are no efficient NIDS capable of detecting various types of recent network attacks, but they can accomplish this by detecting fewer known attack types. Most of the IDS do not address the issue of class imbalance, especially with the CSE-CIC-IDS2018 dataset, which may result in lower detection performance. Certain approaches do not deal with the risk of overfitting and even biases in clustering.

While several IDS exist, traditional methods often fall short in dynamic and high-speed network environments. Modern cyberattacks are fast, sophisticated, and auto-

mated, spreading across networks in seconds. Traditional IDS may take a long time to process and detect attacks, leading to delays in mitigation. Real-time IDS can instantly detect and respond to attacks, preventing damage before it escalates. Many IDSs generate a high number of false positives, overwhelming security teams with excessive alerts.

2.3 Research Objectives

Despite significant progress in anomaly-based NIDS, several challenges remain unresolved. Issues such as feature selection optimization, effective dimensionality reduction, uncertainty handling in clustering, and real-time detection constraints continue to impact IDS performance. Based on the analysis and discussion of the existing state-of-the-art algorithms and the research gaps identified, we have formulated the following objectives for the thesis:

- Review and compare various supervised and unsupervised Machine learning algorithms implemented for intrusion detection in an anomaly-based network environment.
- To investigate feature extraction, selection and ranking approaches during the initial phase for network traffic data.
- Develop a framework for performing dimensionality reduction based on extracted features and further classify the dimensionality-reduced network traffic.
- Develop an anomaly-based Network Intrusion Detection System using Machine Learning model on real-time network traffic flows collected with an aim to detect network attacks.

Chapter 3

Performance Analysis of NIDS using Feature Selection

This chapter presents the analysis of various Machine learning models used for anomaly-based network intrusion detection after selecting the optimal set of features. This study proposes a dynamic feature selection solution based on mutual information and an evolutionary genetic algorithm. The performance analysis demonstrates that the proposed strategy provides a more effective solution for intrusion detection.

3.1 Introduction

Machine learning algorithms have emerged as powerful tools for anomaly-based intrusion detection, enabling the detection of sophisticated attacks. However, the performance of these models is highly dependent on the selection of relevant features that enhance detection accuracy while reducing computational complexity. Many intrusion detection systems also collect network data from various sources that may contain redundant and irrelevant features, leading to an increase in processing time and a low detection rate. Several researchers have also determined that intrusion detection is a classification problem [79, 80, 81]. Another major challenge when it comes to IDS is high-dimensional datasets such as KDD Cup'99, Kyoto 2006+, UNSW-NB15, etc., which interfere with the classification process on the IDS and also have a high computational complexity. Additionally, most feature selection methods overlook feature

dependencies and evaluate each feature in isolation, resulting in lower computational performance than other methods. To the best of our knowledge, most of the existing anomaly-based IDS offer to detect intrusion via supervised machine learning techniques only, whereas unsupervised learning requires classless data. In this field, a comprehensive analysis of supervised and unsupervised machine learning techniques for intrusion detection is needed, particularly after selecting the optimal set of features, which has not been thoroughly explored.

This study introduces a dynamic feature selection approach that leverages mutual information and an evolutionary genetic algorithm, DMI-GA, which considers not only individual features in the feature set but also the degree of association amongst them and the target variable to acquire optimum features. This approach enhances diversity and optimizes feature selection by evaluating relevance and redundancy using a genetic algorithm. The performance and detection accuracy of each selected feature set are evaluated by applying machine learning techniques to these flows and comparing the results to determine whether the performance improves.

The key contributions of the chapter are abstracted as follows:

1. In this study, DMI-GA, a novel hybrid feature selection technique based on mutual information and an evolutionary genetic algorithm, is proposed to obtain an optimal feature set, enabling higher detection accuracy.

2. A novel dynamic fitness function, as the core of the genetic algorithm, is developed to identify optimum features.

3. The efficiency of ML models is investigated against six well-known feature selection methods and the proposed DMI-GA approach, which selects the relevant features from the network traffic feature set to identify the intrusion present.

4. The performance of the proposed feature selection method with the ML classifier is compared and analyzed to distinguish the network attack flows from the benign ones. Furthermore, a statistical analysis was conducted using the Friedman test and a pairwise post-hoc Wilcoxon-Holm test to assess the superiority of the proposed approach over the baseline methods.

This chapter is structured as follows: Section 3.2 presents the methodology and experimental design of the proposed framework, including various feature selection meth-

ods and the proposed feature selection approach in Subsection 3.2.1, as well as the machine learning models in Subsection 3.2.2. The experimental results, a comprehensive evaluation of the proposed approach compared to state-of-the-art methods, and statistical analysis are provided in Section 3.3. Finally, the chapter summary is presented in Section 3.4.

3.2 Proposed Framework

This section describes a novel hybrid feature selection strategy based on mutual information and a genetic algorithm to obtain the optimal set of features from network traffic. This study aims to evaluate the effectiveness of an intrusion detection system in identifying anomalous network attacks. To assess the efficiency of the ML models, this section also presents a performance evaluation using feature sets derived from both state-of-the-art and the proposed feature selection strategies. The proposed model involves three phases: Network traffic collection and pre-processing in the first phase, feature extraction, selection, and ranking in the second phase, and intrusion detection using ML models in the final phase. The proposed framework is represented in Fig. 3.1.

3.2.1 Data Collection and Pre-processing

During the initial phase of the framework, a network traffic packet analyser, Wireshark [82, 83], helped capture the non-malicious or normal network traffic. The intrusion network traffic was imported from the Canadian Institute for Cyber Security's CICDoS2019 dataset [18], the University of Victoria's ISOT Botnet Dataset [17], and the malware capture facility project of the Czech Technical University. The dataset, being in the pcap format, was further pre-processed. For pre-processing, the resulting dataset, originally in raw PCAP format, was first converted to CSV format, having 1,11,53,482 benign network traffic data and 1,03,40,287 malicious traffic instances. Further, this data was cleaned by removing duplicate flow entries in order to reduce redundancy, filling in missing or null values, and removing infinite values by substituting them with the average value of the respective feature. Feature encoding using LabelEncoder was performed to convert the categorical data into numerical form. The ground-truth labels in the datasets are assumed to be correct and representative of their respective attack or

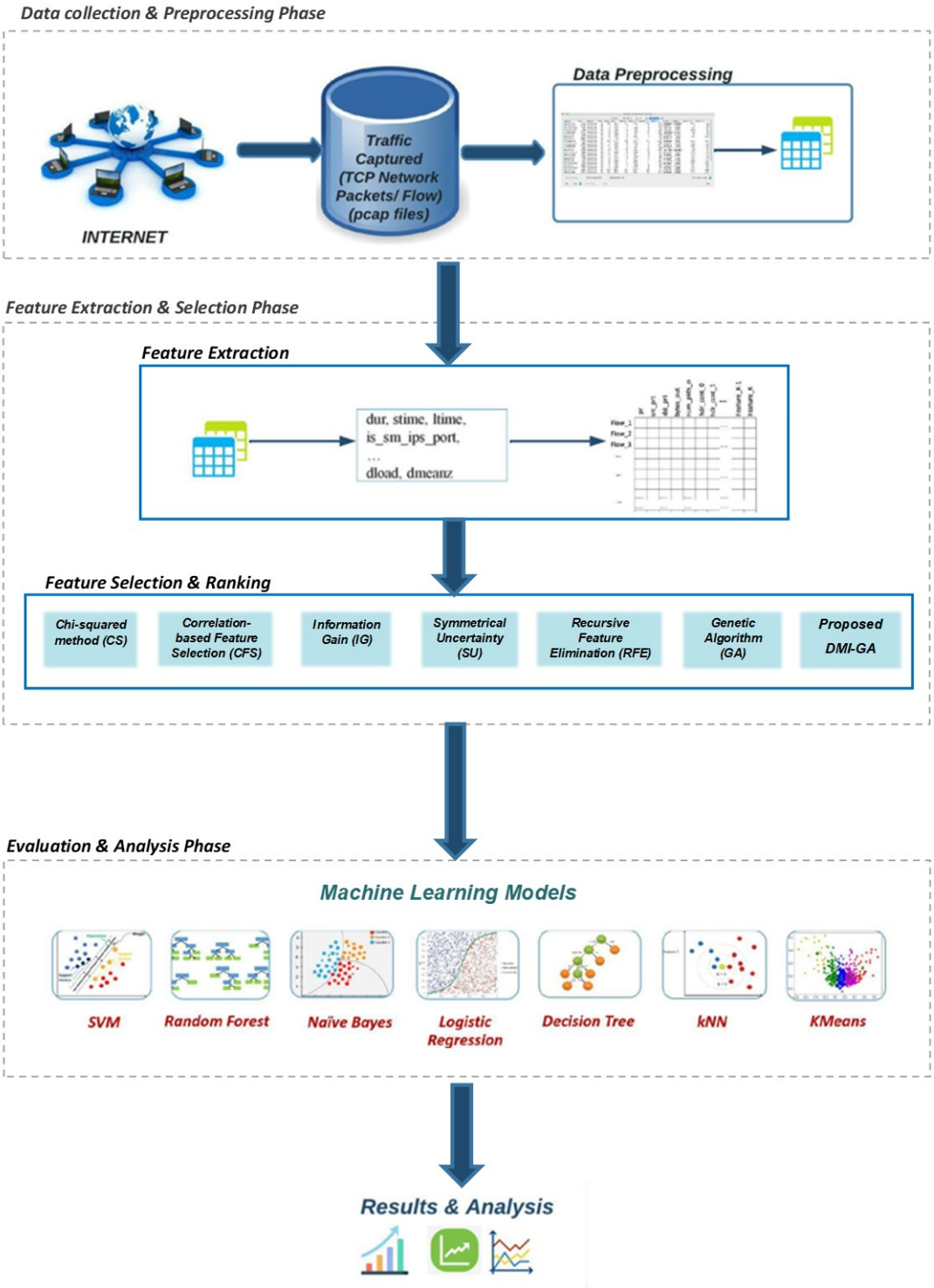

Fig. 3.1: Proposed Anomaly-based NIDS model framework.

benign classes. Likewise, the extracted flow-level features and their preprocessing are assumed to capture meaningful behaviour rather than environment-specific artefacts.

3.2.2 Feature Extraction, Selection, and Ranking

In this study, feature extraction is performed in two stages, extracting 35 statistical features, including conventional and additional features, from network traffic flows. In the first stage, basic statistical and temporal features, such as flow duration, packet count, and average time interval for packets received, were extracted using the statistics feature of the network traffic sniffing tool, Wireshark. In the second stage, behavioural and contextual features, for instance, the number of unique destination IP per n connections, were extracted based on the state-of-the-art models and domain knowledge, as discussed in subsection 2.1.1. Furthermore, additional features were generated from the matched features of the dataset and on the basis of the significant features, as presented in [4], which are vital for anomaly-based intrusion detection for network traffic flows and enabled us to investigate the network more extensively, for example, mean of source packet size. In entirety, these features provide a comprehensive representation of network behaviour, facilitating effective intrusion detection. The extracted features were normalized to a scale of [0, 1]. Table 3.1 briefly describes the extracted features.

Feature selection aims to identify the most relevant attributes while eliminating redundant and irrelevant ones to improve classification performance, enhance processing duration, reduce data and computational complexities, and improve data compatibility with the model. Filter, wrapper, and hybrid or embedded techniques are the three broad categories of Feature Selection strategies [84, 85]. The filter-based approach is chosen for feature selection as it employs statistical methods to rank features and assess their dependency or correlation, thereby identifying an optimized subset. One key advantage of this method is that it prevents overfitting and has low computational requirements. However, it may oversimplify the feature selection process by ignoring the complexities of feature interactions, which can lead to the exclusion of valuable features. The wrapper-based approach, in contrast, evaluates models using a predefined algorithm to determine the optimal feature set. Since it is integrated with a specific algorithm, it generally achieves higher classification accuracy than the filter approach. This method is particularly beneficial for complex models with non-linear relationships

Table 3.1: Network Flow Feature Set

Feature Notation	Feature	Feature Description
F1	srcip	Source's IP address
F2	srcprt	Source's port number
F3	destip	Destination's IP address
F4	destprt	Destination's port number
F5	pkts	Number of packets
F6	bytes	Number of bytes
F7	sdpkts	Number of Source packets to Destination packets
F8	sdbytes	Number of Source bytes to Destination bytes
F9	dspkts	Number of Destination packets to Source packets
F10	dsbytes	Number of Destination bytes to Source bytes
F11	relstart	Relative Start (time duration in sec between start of capture of 1st packet and start of the conversation)
F12	trnsrtsd	Transmission rate Source to Destination (bits/s)
F13	trnsrtds	Transmission rate Destination to Source (bits/s)
F14	avgfld	Average Flow Duration
F15	avgpktsz	Average Packet Size
F16	avgpktszr	Average Packet Size Received
F17	avgpktszs	Average Packet Size Sent
F18	avgtmpktr	Average time between packets received
F19	avgtmpkts	Average time between packets sent
F20	ratioiobyt	Ratio of Incoming bytes to Outgoing bytes
F21	ratioiopkt	Ratio of Incoming packets to Outgoing packets
F22	ratiooibyt	Ratio of Outgoing bytes to Incoming bytes
F23	ratiooipkt	Ratio of Outgoing packets to Incoming packets
F24	meanSpktsz	Mean of source packet size
F25	meanDpktsz	Mean of destination packet size
F26	distSIP	Number of distinctive source IP address per N connections
F27	distDIP	Number of distinctive destination IP address per N connections
F28	minpkts	Minimum time interval between packets sent
F29	maxpkts	Maximum time interval between packets sent
F30	minpktr	Minimum time interval between packets received
F31	maxpktr	Maximum time interval between packets received
F32	connS	Number of connections of same source address
F33	connD	Number of connections to same destination address
F34	fstpktsnt	First packet size sent
F35	fstpktrcd	First packet size received

and intricate feature dependencies. However, its strong coupling with an induction algorithm makes it more time-consuming as it repetitively calls the inducted algorithm to evaluate the subset of features. Hybrid or embedded approaches combine both filter and wrapper techniques, integrating a learning algorithm to optimize feature selection. These methods effectively capture feature interactions and dependencies while leveraging filter-based pre-selection to mitigate the risk of overfitting associated with wrapper methods.

We proposed a hybrid approach, the Dynamic Mutual Information-based Genetic Algorithm (DMI-GA) for feature selection, with the aim of enhancing the performance of machine learning techniques by identifying an optimal set of features. In this study, various other feature selection methods are explored, including filter-based techniques (Chi-Squared, Information Gain, Correlation Feature Selection) [86] and wrapper-based approaches (Recursive Feature Elimination, Genetic Algorithm) [38] for selecting and ranking the optimal set of features.

- **Chi-squared method (CS):** The CS method uses the χ^2 statistic to assess the strength of the connection between each feature and class. The higher the value of χ^2, the higher the dependency between the two events. In terms of feature selection, a feature's occurrence and the class's occurrence are the two events considered. A higher value of χ^2indicates that the two of them, that is, feature and class are dependent. Considering the interdependence of the events, the existence of the feature leads to a higher probability of occurrence of the class. The CS value of a feature is computed as given in Eqn. (3.1):

$$\chi^2 = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{A_i B_j}{N}. \tag{3.1}$$

  where, the number of classes is m, n is the number of intervals, N is the total number of instances, A_i is the number of instances in the i_{th} interval, B_j are the number of instances in the j_{th} class.

- **Correlation Feature Selection Measure (CFS):** Correlation refers to the degree to which two features are in a linear relationship with each other. It evaluates the similarity between the two features. The correlation coefficient of two linearly dependent features is ± 1, while the uncorrelated features have a correlation

coefficient of 0. With respect to a correlation-based evaluation function, CFS, an elementary filter algorithm, ranks the subsets of features. A low correlation feature associated with the class should be disregarded as they are considered irrelevant features. Eqn. (3.2) defines the CFS's feature subset evaluation function [87] as:

$$M_F = \frac{k\overline{r}_{cf}}{\sqrt{k + k(k-1)\overline{r}_{ff}}}. \tag{3.2}$$

where, M_F is the "merit" of a feature subset F with k features, $\overline{r}_{cf}$ is the mean of correlation between feature and class ($f \in S$), and $\overline{r}_{ff}$ is the average inter-correlation between two features. The numerator specifies how predictive a feature set in a class is, and the denominator depicts the presence of redundancy amongst the features.

- **Information Gain (IG):** Entropy is a metric for assessing disorder or uncertainty in a system [88]. A system with high entropy would be unpredictable and more disordered. Thus, it measures how unpredictable a data distribution is. Consider nominal valued features, Y comprising of the individual probabilities of the values $y \in Y$. The entropy of Y, $H(Y)$, is evaluated as shown in Eqn. (3.3).

$$H(Y) = -\sum_{y \in Y} p(y) \log_2(p(y)). \tag{3.3}$$

Suppose in the training data, the observed values of Y are segregated with respect to another feature X's values, and the entropy of Y in association with the partitions brought by X is less than the entropy of Y before partitioning. In that case, an association or an affiliation among the features X and Y exist. Eqn. (3.4) represents the entropy of Y after observing X.

$$H(Y|X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) \log_2(p(y|x)). \tag{3.4}$$

Information Gain is the extent to which the entropy of Y decreases and represents the additional information about Y that X provides [89], which is represented by Eqn. (3.5).

$$IG(Y|X) = H(Y) - H(Y|X). \tag{3.5}$$

Thus, X is considered to be more correlated to Y than to some feature Z belonging to the same feature set as X and Y, iff $IG(Y|X) > IG(Z|X)$.

- **Symmetric Uncertainty (SU):** Symmetry is an essential asset for measuring the intercorrelation between two features. Symmetrical Uncertainty overcomes the Information Gain's inherent bias in favoring features or attributes with more values by normalizing its value to the range of [0, 1].

$$SU(X,Y) = 2 \times \frac{IG(X,Y)}{H(X) + H(Y)}. \tag{3.6}$$

Features with the higher value of SU are ranked higher than the other features. SU's maximum value, SU = 1 shows that X and Y are highly correlated and SU = 0 depicts that X and Y are uncorrelated. Eqn. (3.6) describes how SU is evaluated and the relationship between SU and IG.

- **Recursive Feature Elimination (RFE):** Recursive Feature Elimination [90] is a wrapper-based feature selection technique. As the name suggests, it recursively eliminates the least important feature, one at a time, evaluating the ML model's performance at every iteration. It starts with all features, ranks them by their relevance to the target variable, and then removes the least important feature(s), repeating this process until the desired model performance is achieved. RFE considers interactions among features during the elimination process, which can lead to selecting more informative subsets of features.

- **Genetic Algorithm (GA):** Genetic algorithm [91] is a search heuristic inspired by the process of natural selection based on "Survival of the fittest". It starts with a population of potential solutions encoded as chromosomes. Through iterative processes of selection, crossover, and mutation, GAs evolve solutions based on their fitness scores. They are advantageous for feature selection as they efficiently explore large search spaces to identify the most relevant features, improving model performance and reducing dimensionality.

- **Proposed Dynamic Mutual Information-based Genetic Algorithm (DMI-GA):** The proposed feature selection approach, DMI-GA, is an amalgamation and enhanced version of mutual information-based feature selection (MIFS) [92] and genetic algorithm [93]. As the optimization strategy employed to find an optimal feature set, GA helps evolve the population of feature subsets, leveraging the principles of natural selection through selection, mutation, and crossover operations [94]. This introduces stochasticity and diversity, helping to explore the search space and thoroughly avoid local optima. Mutual information (MI) is a fundamental concept in information theory used to gauge how much information one random variable provides about another. In feature selection, it evaluates the relationship between each feature and the target variable, finding features that offer significant predictive insight. Its capability to handle nonlinear and intricate relationships between variables makes MI a crucial asset in enhancing the accuracy of models by selecting the most relevant features. Mathematically, MI is computed as:

$$I(A;B) = \sum_{i=1}^{m}\sum_{j=1}^{n} p(a_i, b_j) \log\left(\frac{p(a,b)}{p(a)p(b)}\right) \tag{3.7}$$

where, $I(A;B)$ denotes the mutual information between a random feature, say, A and the target variable B. It measures how much information A provides about B, helping to determine the relevance of A for further prediction. $p(a,b)$ is the joint probability of A and B. $p(a)$ and $p(b)$ are the marginal probability of A and B, respectively.

For this proposed feature selection solution, we integrate the filter method to evaluate the mutual information with the genetic algorithm wrapper method to create a hybrid approach, DMI-GA. Fig. 3.2 presents a flowchart of DMI-GA that begins with initializing the population with a random set of solutions or feature subsets and further evaluating the initial population. In the next step, we evaluate the fitness of all individuals in the initial population with the proposed novel fitness function. To evaluate this fitness function, the mutual information between each feature F and each target is determined. Along with MI, an adaptive trade-off parameter (λ) between relevance and redundancy is dynamically adjusted based

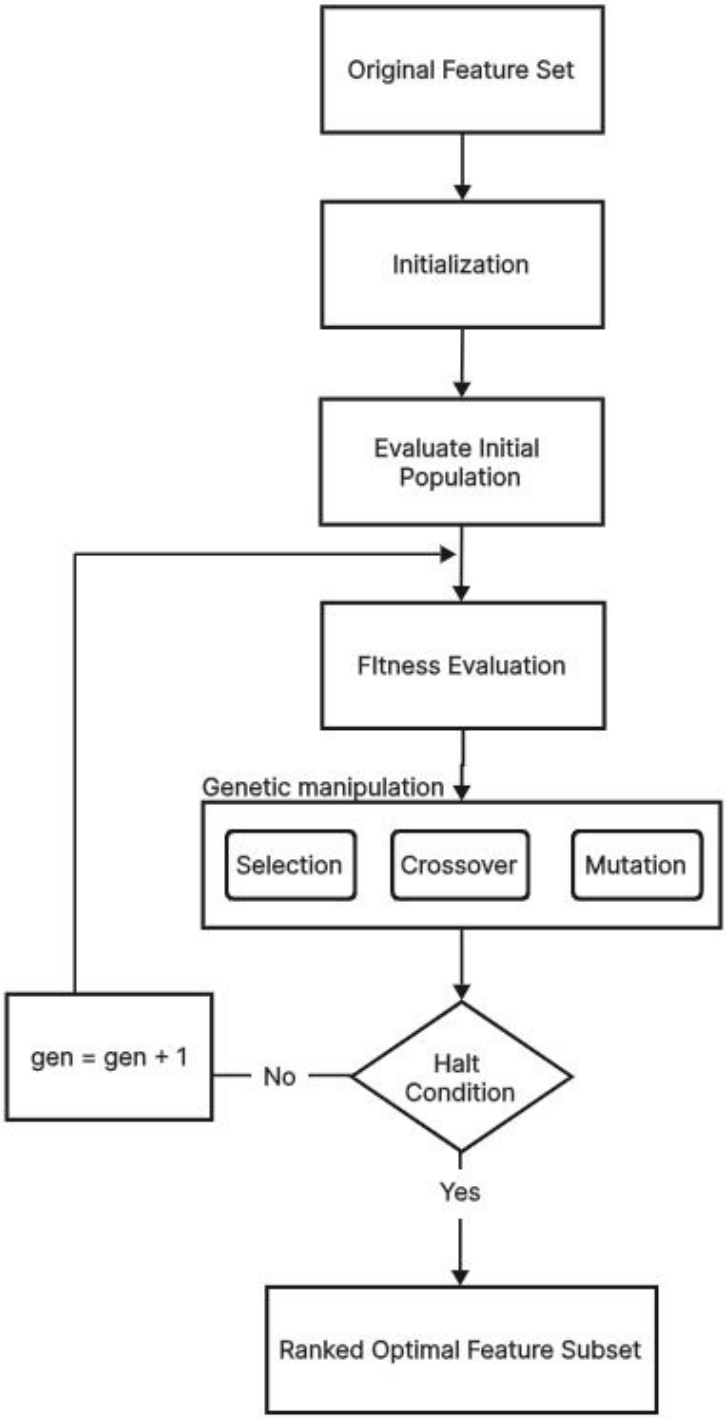

Fig. 3.2: Proposed Dynamic Mutual Information-based Genetic Algorithm (DMI-GA) for Feature Selection

on the population evolution instead of a fixed or heuristically determined trade-off parameter. This allows the algorithm to adaptively balance the importance of relevance and redundancy during the optimization process. With the MI of the population and the adaptive λ, the fitness function is evaluated as:

$$Fitness = \sum_{i=1}^{n} I(A_i; B) - \lambda(t) \sum_{i=1}^{n} \sum_{j=1}^{n} I(A_i; A_j) \tag{3.8}$$

where, the adaptive λ(t) function dynamically adjusts λ based on the generation counter t, and is determined by:

$$\lambda(t) = \lambda_0 * \left(1 - \frac{t}{max\_gen}\right) \tag{3.9}$$

where, λ_0 is the initial λ value and max_gen is the maximum number of generations.

In the next phase, three vital GA operators, selection, crossover, and mutation, are performed to produce the optimal feature set. Initially, two individual parents are selected from the current generation or population to produce the next generation. This selection is done either randomly or via selection methods such as Tournament selection, Roulette wheel selection, Stochastic Universal Sampling, etc. We have opted for Tournament Selection in our proposed feature selection strategy as it is capable of working with negative fitness values as well. Crossover combines the selected two parents to create offspring, allowing the mixing of feature subsets and potentially discovering better combinations. By randomly flipping bits in a feature derived from the crossover, mutation introduces diversity into the population, helping to explore new feature subsets that might have been overlooked. At the end of each generation, the population is assessed to determine if the algorithm should terminate. If the termination condition is not satisfied, the population undergoes re-evaluation through fitness function calculation and the application of GA operators. This cycle continues until the stopping criteria are fulfilled. The termination condition is met on the convergence of the fitness function.

The different phases involved in identifying the optimal set of features using the proposed DMI-GA feature selection method are detailed in Algorithm 3.1.

Algorithm 3.1 Proposed DMI-GA feature selection method

Input: Original feature set $F = \{f_1, f_2, \ldots, f_n\}$ with population size N; maximum number of generations $g_{\max}$; adaptive trade-off parameter λ.
Output: Optimal feature subset
procedure DMI-GA
    $S \leftarrow \emptyset$
    Initialize generation counter $g \leftarrow 0$
    Initialize a population of i individuals with $P_0 \leftarrow$ random
    while $g \leq g_{\max}$ **do**
        Compute mutual_info $MI_i = I(A_i; B)$ by Eqn. (3.7)
        Repeat
        Compute fitness value for every i^{th} individual by Eqn. (3.8)
        for $i = 1$ to n **do**
            Select two individual parents via Tournament Selection
            Perform crossover operation on selected parents to generate offspring
            Perform mutation on generated offspring from Step 13 to form new offspring, R_i
            Replace $P_i \leftarrow R_i$
        end for
        $g \leftarrow g + 1$
        until $Convergence[\lambda]$
    end while
    Rank f_i in P_i in descending order based on MI_i
    $S \leftarrow P_i$
    return S
end procedure

3.2.3 Machine Learning Models for Intrusion Detection

Machine learning models play a crucial role in binary classification by identifying network traffic as either benign (or normal) or intrusive based on selected features. The selected features obtained from each of the feature selection methods were fed to the state-of-the-art ML models, including logistic regression, decision tree, naive bayes, *k*-nearest neighbour, random forest, SVM, and K-means. The description for the ML models employed in this study is presented in subsection 1.1.2.

This study considered the seven mentioned state-of-the-art ML models for the experiments. LR can efficiently process large-scale network traffic datasets, ensuring quick and accurate anomaly detection, and provides probability scores for classification, which helps in fine-tuning detection thresholds and reducing false alarms in network security [95]. We chose the DT classifier as it is simple to interpret, is able to handle complex problems, and is capable of handling non-linear relationships amongst the feature set. The NB classifier performs well even with high-dimensional network traffic data, which is common in IDS, and can handle both discrete and continuous network features [96]. Due to the ease of use for *k*-NN, its capability to handle both categorical and numerical data, and lack of pre-requisite assumptions about the underlying data distribution, we have opted for this flexible model to classify network attacks from benign ones. As RF performs efficiently for high-dimensionality datasets, takes less training time in comparison to other classifiers, and addresses the issue of overfitting by making predictions based on either majority voting or averaging [97], it is therefore employed in this work. SVM performs well even when network traffic data has a large number of features, making it suitable for complex intrusion detection datasets. Also, SVM maximizes the margin between different classes, improving its ability to correctly classify previously unseen network traffic patterns [98].

3.3 Experimental Results and Analysis

This section discusses the outcomes from various stages of the proposed IDS framework. By implementing machine learning algorithms on the extracted, selected, and ranked feature sets, we compare and analyze the results based on the performance evaluation metrics discussed in the previous section. We executed two experiments on this

dataset. In the first experiment, ML models used the original pre-processed dataset, while in the second experiment, ML techniques performed intrusion detection on the optimal feature set obtained after feature selection.

The experiments were performed on a computer running the Windows 10 operating system, equipped with 16GB RAM and an i7 (9^{th} generation) processor. The entire experiment was conducted using Spyder (64-bit). The feature selection methods outlined in subsection 3.2.2 were precisely implemented, and the optimal ranked feature set was extracted from the original features, as presented in Table 3.1, for further analysis.

3.3.1 Hyperparameter Tuning and Validation

The process of tuning hyperparameters for both the proposed DMI-GA feature selection method and the ML classifiers, as well as the procedures employed to assess robustness, is discussed in this subsection. All model selection decisions were based on validation accuracy. The aim is to make sure that improvements in performance come from sound model selection rather than favourable randomness or evaluation bias. Consequently, we compared six well-known feature selection approaches to evaluate the relevance of all features in the feature set, independently of the classifier.

Hyperparameters for the proposed DMI-GA were tuned with respect to population size, maximum generations, crossover rate, mutation rate, and the seven ML models used for intrusion detection. This tuning was carried out to ensure the optimal performance of both the DMI-GA feature selection method and the subsequent ML classifiers. The process was aimed at identifying parameter configurations that maximize classification accuracy, reduce overfitting, and maintain computational efficiency.

The DMI-GA parameters, population size, maximum number of generations, crossover probability, and mutation probability were initially selected from common ranges used in the literature on evolutionary feature selection. Then, these were fine-tuned through pilot experiments in which each parameter was varied independently, while others remained constant, to assess its impact on two criteria, classification accuracy using selected features and convergence rate in terms of generations required. During tuning, the population size was varied from 50 to 150 and the number of generations from 0 to 160, each with a step size of 20. The crossover probability ranged from 0.70 to 0.90 with a step size of 0.05, while the mutation probability varied from 0.01 to 0.10 with a

step size of 0.02.

Following feature selection, the machine learning models were tuned individually using 10-fold cross-validation on the training set to ensure fair comparison. Table 3.2 defines the typical parameter ranges for each ML technique and the selected value.

Table 3.2: Hyperparameter tuning ranges and optimal settings for six ML techniques.

ML Technique	**Parameter Tuned**	**Tested Range**	**Selected Value**
LR	Regularization *(R)*	*R*: 0.1-10	1.0
DT	Max. Tree Depth *(d)*	*d*: 5-25	20
NB	Smoothing *(α)*	α: 0.1-1.5	1.0
kNN	No. of Neighbours *(k)*	*k*: 1-15	7
RF	Number of Trees *(n)* Max. Depth *(D)*	*n*: 50-300 *D*: 5-20	*n* = 200, *D* = 15
SVM	Penalty *(C)* Stopping criteria tolerance *(t)*	*C*: 0.5-15 *t*: 0.001-0.01	*C* = 11; *t* = 0.001
K-Means	No. of Clusters *(x)* Convergence tolerance *(cv_tol)*	*x*: 2 *cv_tol*: 1e-3, 1e-4	*x* = 2; *cv_tol* = 1e-4

Table 3.3 summarizes the parameters and their values used for the proposed DMI-GA feature selection method. These parameters are crucial for controlling the evolutionary process and ensuring efficient convergence toward an optimal feature subset.

The population size is set to 100, ensuring a balance between genetic diversity and computational efficiency. The algorithm runs for a maximum of 150 generations, preventing excessive computations while allowing sufficient iterations for convergence. A crossover probability of 0.85 promotes diversity by exchanging genetic material between parent solutions, while a mutation probability of 0.05 introduces small random variations to explore new solutions without disrupting good feature combinations. The initial trade-off parameter, $\lambda_0 = 1.0$ helps dynamically balance feature relevance and redundancy based on mutual information. To ensure stability, the convergence tolerance is set to 1×10^{-5}, stopping the algorithm when improvements become negligible. Finally, tournament selection is employed as the selection method, ensuring that stronger

Table 3.3: Parameter setting for proposed DMI-GA feature selection method

Parameter	Value
Population size (pop_{size})	100
Maximum number of generations (gen_{max})	150
Probability of crossover ($Crossover\_rate$)	0.85
Probability of mutation ($Mutation\_rate$)	0.05
Initial trade-off (λ_0)	1.0
Convergence tolerance (tol)	1×10^{-5}
Selection method	Tournament selection

candidates have a higher chance of survival while maintaining diversity within the population. These parameter choices collectively enhance the efficiency and effectiveness of the proposed feature selection approach.

We quantified how DMI-GA performance varies with the crossover–mutation operators using three reference classifiers, RF, SVM, and kNN. Fig. 3.3 shows the validation accuracy heatmaps for each evaluator classifier. All three heatmaps exhibit a shared optimum region centered at crossover ≈ 0.85 and mutation ≈ 0.05, with a robust plateau spanning 0.80–0.90 crossover probability and 0.03–0.07 mutation rate. Minor variation in the peak, for instance, a slightly wider plateau with kNN, does not affect the overall optimum choice. This indicates that operator selection is not limited to any one classifier and supports the configuration presented in Table 3.3.

Fig. 3.4 shows the convergence behaviour of the proposed DMI-GA method with the three reference classifier evaluators, RF, kNN, and SVM. It can be observed that for the random forest setting, the validation accuracy increases sharply within the first 40 generations, reaching $\approx 97\%$. After that, it gradually stabilizes and converges around 99.90%. This shows that with RF as the evaluator, the DMI-GA quickly finds high-quality feature subsets and converges early. Similar to RF, the curve rises rapidly for the kNN evaluator and converges around 99.7%–99.8%. The SVM-based convergence curve starts slower but increases quickly, converging around 99.60%. SVM converges more slowly than RF and kNN but achieves comparable accuracy.

For all three classifiers, performance improves smoothly and stabilizes by ≈approximately 150 generations, justifying $gen_{max} = 150$ as a balanced performance setting.

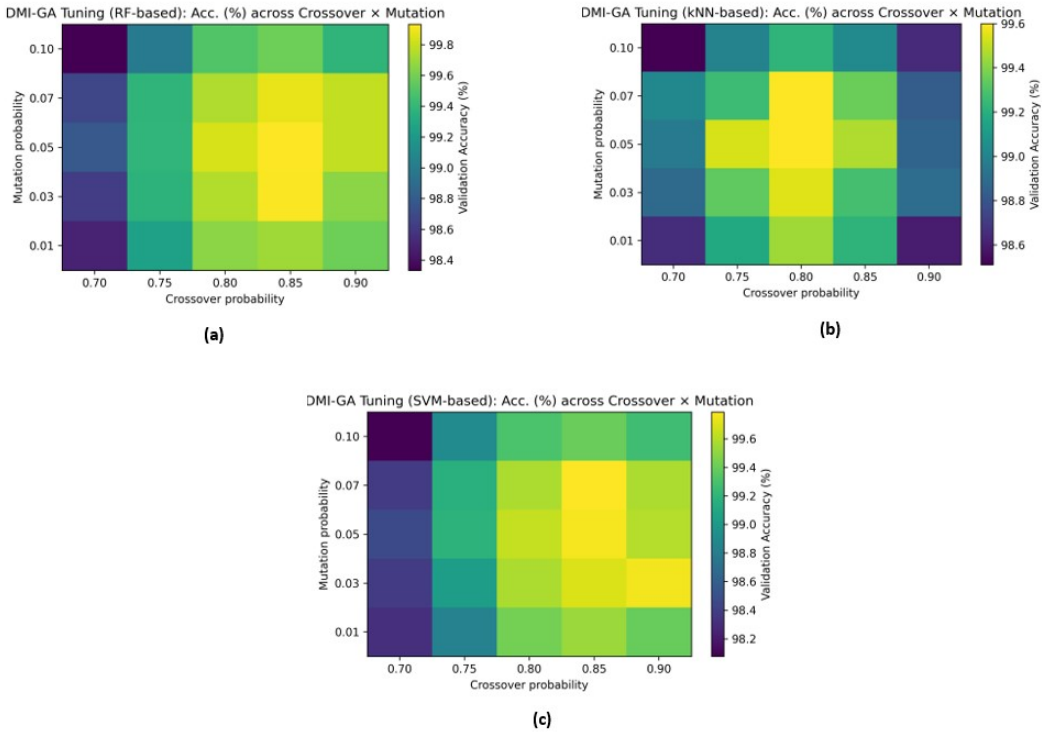

Fig. 3.3: Validation accuracy (%) across crossover and mutation probabilities for DMI-GA parameter tuning with (a) RF, (b) kNN, and (c) SVM

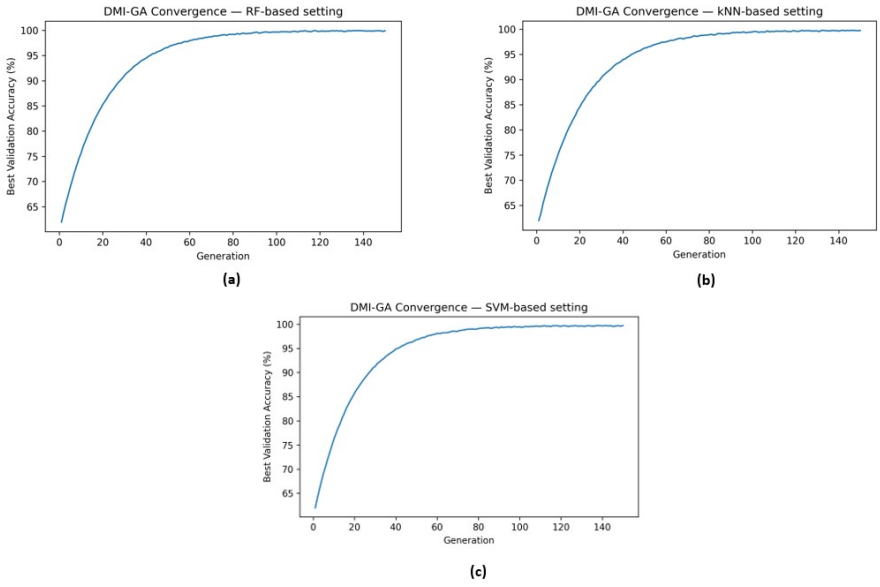

Fig. 3.4: DMI-GA convergence curves showing validation accuracy across generations for DMI-GA with (a) RF, (b) kNN, and (c) SVM evaluators

To demonstrate the systematic tuning of the ML models, each classifier was tuned with 10-fold cross-validation using accuracy. To keep the explanation concise, yet transparent, we present three representative tuning landscapes with RF, SVM, and kNN, as visualised in Fig. 3.5.

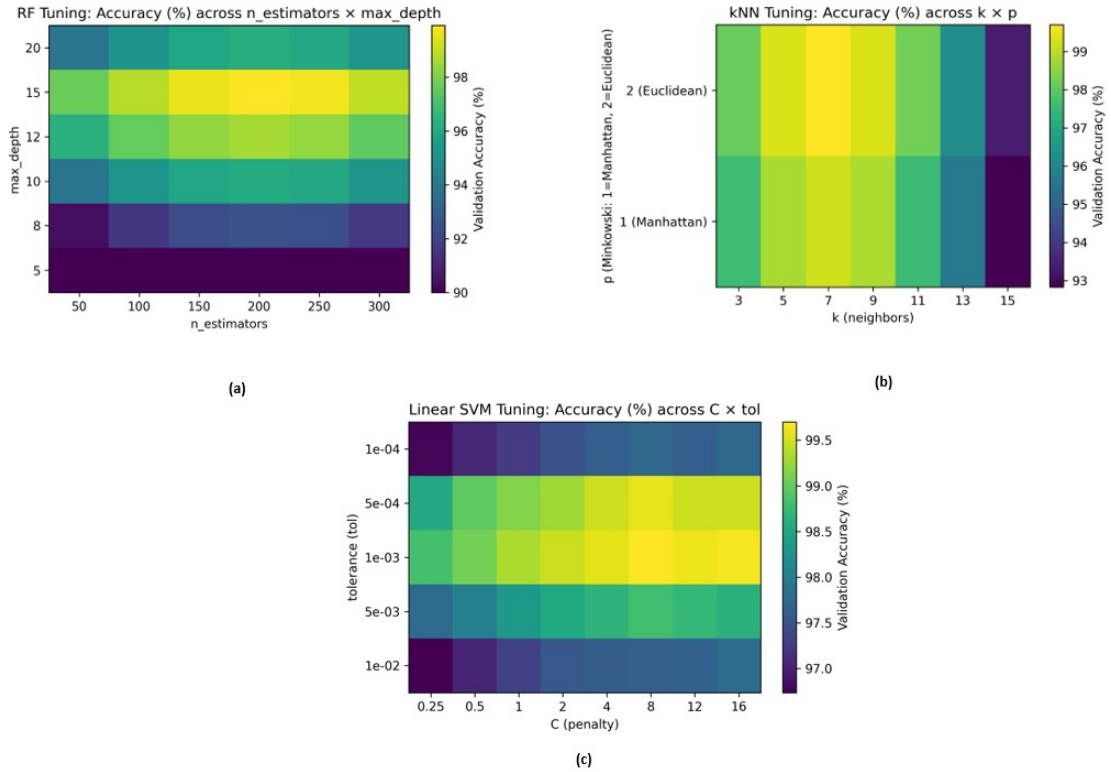

Fig. 3.5: ML Classifier hyperparameter tuning for (a) RF, (b) kNN, and (c) SVM

It was observed that accuracy in random forest monotonically improves as trees increase, then saturates beyond 200 trees. Depth up to approximately 15, after which it gains a plateau or slightly dips. For kNN, the number of neighbours (k) varied from 3-15, and the distance metrics were tested for the Euclidean and Manhattan distance. Accuracy rises from small k to a peak around $k \approx 7$, then declines for larger k. This happens because very small k produces high-variance, noisy decision boundaries and very large k oversmoothens class boundaries and blends minority attack points with normal traffic. Manhattan (p=*1*) is consistently slightly worse than Euclidean (p=*2*) across the grid. In the linear SVM heatmap, accuracy forms a clear ridge along the penalty C, with the best region near $C \approx 11$ and tolerance around 0.001. Too small a C or a

weaker penalty leads to underfitting with overly wide margins, while excessively large *C* risks overfitting by enforcing hard margins. Similarly, very loose tolerances prevent proper convergence, while overly tight tolerances add unnecessary computational burden without accuracy gain. Thus, the setting $C = 11$ and $tol = 0.001$ achieves a balanced trade-off, resulting in stable and near-optimal validation accuracy.

These example landscapes illustrate the selection process used across all models. The final chosen values per ML techniques are summarized in the hyperparameter table 3.2.

Table 3.4 presents the optimal selected subset of features along with their ranking, obtained from the original dataset of 35 features by various feature selection methods (FS), along with the proposed DMI-GA employed. The features were fed iteratively to the classifier in the order of ranks provided by the respective feature selection methods. In addition, features are fed incrementally according to their rank in the classifier, and the best performance is recorded in terms of the metrics stated in subsection 3.3.2.

This study aims to analyse and comprehensively compare different combinations of state-of-the-art feature selection methods and the proposed hybrid feature selection method DMI-GA with ML algorithms for anomaly-based network intrusion detection. We also analyse the results when the original dataset is under consideration, that is, with no feature selection (No FS). Taking into account a combination of No FS along with seven feature selection methods with seven ML algorithms, considering the value of k in kNN as $k = 5$, $k = 10$, and $k = 15$, 72 models were experimented with for intrusion detection

3.3.2 Performance Evaluation Metrics

To assess the effectiveness of anomaly-based network intrusion detection systems, various performance metrics are used. These metrics help quantify the system's ability to accurately distinguish between normal and anomalous network traffic. This study considers Accuracy, Precision, Recall, and F1-score as primary criteria for evaluating the performance [99]. Accuracy measures the overall correctness of the system. Precision, on the other hand, indicates how many of the detected anomalies were actual intrusions, helping to minimise false alarms. Recall, also called sensitivity, measures how well the system identifies all intrusions, focusing on minimising missed attacks. F1-score bal-

Table 3.4: Selected and Ranked Optimal Feature Set based on Various Feature Selection Algorithms

Feature Selection Method	#Features Selected	Selected Features and their Ranking
CS	28	*F22, F23, F16, F5, F35, F25, F12, F24, F9, F31, F17, F8, F34, F10, F6, F30, F11, F15, F13, F4, F18, F14, F28, F29, F19, F20, F7, F2*
CFS	17	*F9, F5, F29, F4, F7, F15, F6, F30, F17, F16, F13, F28, F10, F31, F12, F2, F8*
IG	21	*F11, F12, F2, F10, F35, F20, F32, F30, F24, F16, F29, F28, F6, F4, F17, F33, F8, F25, F34, F15, F31*
SU	21	*F9, F15, F28, F25, F17, F2, F6, F34, F7, F16, F10, F5, F14, F18, F31, F24, F19, F35, F8, F4, F11*
RFE	14	*F5, F2, F10, F16, F34, F29, F28, F4, F6, F17, F30, F8, F31, F15*
GA	15	*F12, F13, F4, F6, F2, F15, F16, F17, F5, F7, F8, F28, F30, F25, F24*
Proposed DMI-GA	18	*F2, F4, F6, F11, F8, F5, F7, F24, F25, F15, F17, F16, F12, F13, F30, F31, F29, F28*

ances precision and recall, providing a single measure of detection effectiveness. The classifier parameters were considered optimal if they produced the highest metric values.

To statistically validate the model's efficiency to an independent set of features, we performed two widely popular cross-validation strategies, K-fold (for $K = 10$) and Leave-one-out cross-validation (LOOCV) strategies [100]. Cross-validation offers a more accurate assessment of a model's ability to generalize, as it evaluates the model on various validation sets. This process helps to mitigate overfitting by offering a reliable estimate of how well a model will perform on data it has not encountered before. The 10-fold cross-validation was selected as it provides a reliable bias-variance balance with manageable computation, while LOOCV offers the most exhaustive use of data for validation, though at the expense of heavier computational overhead. Including both strategies allowed us to compare the stability of results under different levels of data

reuse. In both CV schemes, stratification was applied to preserve the original class distribution of normal and attack instances within each fold, ensuring balanced evaluation. The accuracy and F1-score values of K-fold cross-validation (K = 10 and LOOCV) of a model are obtained by averaging them across each fold of k-fold cross-validation.

3.3.3 Performance Comparison of Combinations of ML and FS Techniques

This subsection introspects the performance of the eight FS methods in conjunction with the seven ML models in terms of intrusion detection accuracy (%) as presented in Table 3.5.

The feature selection methods were observed to significantly improve the detection accuracy as compared to without feature selection, that is, No FS. No FS method gave the lowest accuracy, whereas FS methods like CFS, RFE, GA, and the proposed DMI-GA significantly enhanced the accuracy. Additionally, CFS, IG, and SU also improve accuracy but are less effective than GA-based methods. The proposed DMI-GA FS method consistently outperformed all others, achieving the highest detection accuracy across all ML classifiers. Also, RF performed the best among ML models across different FS methods, reaching a maximum accuracy of 99.94% with the proposed DMI-GA. Fig. 3.6 is a visualisation of the detection accuracy of each of the ML models, demonstrating that in every aspect of the feature selection, the proposed DMI-GA feature selection strategy outperformed the others, and NB presented the weakest performance across all FS methods.

The bold values across the experimental results presented emphasize the most significant value(s) compared to the rest.

Tables 3.6 and 3.7 represent the precision and recall values, respectively, for all models combining ML and FS strategies. In terms of these metrics, again, it can be observed that DMI-GA with RF has the highest precision of 96.47% and the highest recall of 99.99%. No FS resulted in lower precision which indicates more false positives. It is also noted that certain methods, such as SU and CFS, achieved high recall but lower precision, suggesting they identify a large number of anomalies but with an increased rate of false positives.

Table 3.5: Detection Accuracy (%) for Combinations of ML and FS methods

Feature Selection (FS)	LR	DT	NB	kNN			RF	SVM	K-Means
				k=5	k=10	k=15			
No FS	94.62	95.68	89.17	94.99	94.92	94.88	95.79	96.11	93.24
CS	97.18	99.86	91.23	99.56	99.44	99.17	99.87	99.10	96.96
CFS	96.08	96.12	90.09	96.29	96.17	96.09	96.44	96.13	95.77
IG	96.36	98.99	90.15	98.71	98.69	98.55	98.98	98.79	96.88
SU	96.23	97.98	89.99	97.32	97.18	97.22	98.02	97.47	96.43
RFE	97.15	99.76	93.44	99.20	98.89	98.73	99.85	98.82	96.91
GA	97.99	99.87	94.00	99.58	99.49	99.22	99.88	99.18	97.27
Proposed DMI-GA	98.10	99.89	94.97	99.77	99.71	99.65	**99.94**	99.70	97.88

Fig. 3.6: Comparison of Detection Accuracy (%) of ML models with the FS methods

Table 3.6: Precision (%) for Combinations of ML and FS methods

Precision (%)									
Feature Selection (FS)	LR	DT	NB	kNN			RF	SVM	K-Means
				k=5	k=10	k=15			
No FS	90.16	91.38	85.99	90.88	90.75	90.56	91.38	92.27	89.79
CS	93.99	95.68	88.73	95.47	95.44	95.29	95.71	95.14	93.64
CFS	92.15	92.31	86.69	92.61	92.60	92.54	92.89	92.49	91.92
IG	93.32	95.42	87.18	95.39	95.26	95.16	95.68	95.10	93.60
SU	93.30	95.16	86.81	94.61	94.57	94.43	95.16	94.61	93.27
RFE	93.89	95.59	88.65	95.43	95.38	95.20	95.70	95.12	93.62
GA	94.78	95.97	89.69	95.77	95.48	95.33	96.10	95.29	93.71
Proposed DMI-GA	95.02	96.39	90.86	96.18	95.99	95.71	**96.47**	95.66	93.99

Table 3.7: Recall (%) for Combinations of ML and FS methods

Recall (%)									
Feature Selection (FS)	LR	DT	NB	kNN			RF	SVM	K-Means
				k=5	k=10	k=15			
No FS	96.75	97.86	91.43	96.89	96.84	96.81	97.88	98.99	96.09
CS	99.78	**99.99**	97.55	99.86	99.84	99.84	**99.99**	99.82	98.91
CFS	98.04	98.10	93.79	98.19	98.18	98.15	98.28	98.14	97.78
IG	98.79	99.90	97.49	99.78	99.66	99.52	99.92	99.43	98.89
SU	98.78	99.47	95.68	99.39	99.38	99.38	99.49	99.39	98.68
RFE	99.52	99.92	97.51	99.80	99.66	99.61	99.95	99.59	98.89
GA	99.79	99.98	97.93	99.89	99.84	99.84	99.98	99.84	98.95
Proposed DMI-GA	99.80	**99.99**	98.26	99.90	99.87	99.86	**99.99**	99.85	98.99

This trend is clearly visible in Fig. 3.7. Across all ML techniques, the NoFS method had the highest false positive rate (FPR) and an elevated false negative rate (FNR), resulting in the model overwhelming administrators with spurious alerts while still failing to detect real intrusions. Without feature selection, IDS generates excessive false alarms, with FPR peaking at nearly 13%, which is impractical for real-world deployment. Considering from NoFS to DMI-GA, shows a huge FPR drop from approximately 8-10% down to approximately 3-4%, while FNR to well below 1%, approximately 0.1% to 0.5%. This corresponds to a 40% to 60% reduction in false alarms and an 85–95% reduction in missed attacks across ML models. For instance, with the RF–DMI-GA, the FPR is only 3.39%, indicating that roughly 1 in every 30 benign flows is incorrectly

flagged as malicious, while the FNR is as low as 0.01%, missing only about 1,034 attacks out of more than ten million, which suggests almost complete attack detection. Overall, DMI-GA consistently minimizes both false alarms and missed intrusions, regardless of the classifier employed. Even with inherently strong models such as RF and SVM, DMI-GA still provides a measurable operational benefit by generating fewer alerts without compromising recall. These results confirm that DMI-GA delivers the most practical feature subset for deployment, giving the right balance between strong detection capability and efficient operation.

After these experiments, the observations suggested that combining machine learning techniques with feature selection for intrusion detection is more efficient and accurate when using the optimal set of features in comparison to the original set of features.

3.3.4 Enhanced Performance Comparison of ML and FS Techniques with Cross-Validation

This subsection presents the performance evaluation of various ML models combined with different FS techniques, incorporating cross-validation to ensure robustness and generalizability. The results offer valuable insights into how effectively the models perform across various data splits, ensuring a more comprehensive evaluation of the intrusion detection system.

Table 3.8 highlights the performance of the combinations of ML and FS techniques based on the detection accuracy achieved and the F1 score. As observed in Table 3.5, the k-NN classifier achieves optimal performance for intrusion detection at k=5. Therefore, for cross-validation, we conducted experiments exclusively at k=5 for this model.

We observed that the LOOCV strategy provides more robust results than the 10-fold CV, suggesting its robustness in evaluating model performance across different data subsets. The results also suggested that the proposed DMI-GA method consistently achieves the highest accuracy and F1-score across all ML models. The highest accuracy of 99.91% (LOOCV) and 99.61% (10-Fold CV) is observed for the Random Forest classifier using DMI-GA, demonstrating its superiority in selecting relevant features. Additionally, among all the FS techniques, GA performs more efficiently but is slightly outperformed by DMI-GA, indicating that the additional mutual information-based enhancement improves the feature selection effectiveness. As declared previously, the No

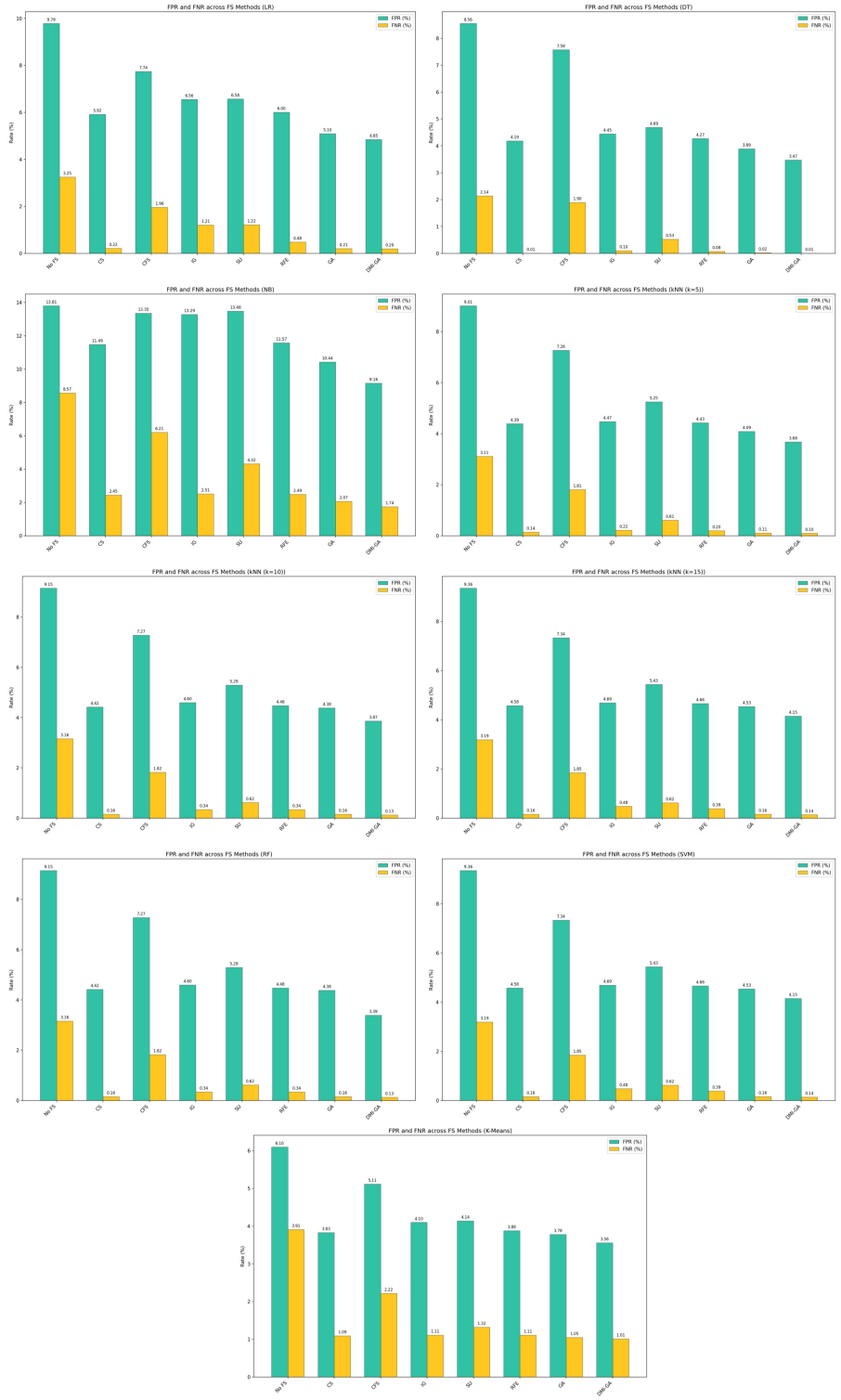

Fig. 3.7: Comparison of FPR (%) and FNR (%) across nine ML models and eight FS methods

Table 3.8: Performance analysis of ML classifiers with the FS methods using two cross-validation strategies

Classifier	FS method	Accuracy (%)		F1-Score (%)	
		10-Fold CV	LOOCV	10-Fold CV	LOOCV
LR	NoFS	94.33	94.50	93.62	94.24
	CS	96.99	97.04	96.66	96.70
	CFS	95.71	95.86	94.68	94.90
	IG	95.90	96.18	95.80	95.88
	SU	95.83	96.11	95.79	95.86
	RFE	96.87	96.99	96.41	96.52
	GA	97.88	97.92	96.94	97.12
	DMI-GA	97.96	98.00	97.20	97.25
DT	NoFS	94.65	95.10	94.00	94.41
	CS	99.37	99.68	97.09	97.69
	CFS	95.66	96.01	94.83	95.02
	IG	98.41	98.65	96.59	97.51
	SU	97.44	97.80	96.37	97.17
	RFE	99.09	99.42	96.85	97.61
	GA	99.43	99.71	97.33	97.83
	DMI-GA	99.52	99.79	97.60	98.06
NB	NoFS	88.99	89.06	88.41	88.53
	CS	91.08	91.17	92.79	92.83
	CFS	89.65	89.82	89.86	90.00
	IG	89.97	90.05	91.89	91.95
	SU	89.73	89.82	90.90	90.93
	RFE	90.86	90.94	92.70	92.77
	GA	93.89	93.94	93.49	93.53
	DMI-GA	94.80	94.88	94.11	94.32
k-NN (k=5)	NoFS	94.57	94.63	93.88	93.69
	CS	99.10	99.33	97.06	97.52
	CFS	95.52	95.99	94.81	95.00
	IG	98.36	98.50	96.43	97.44
	SU	96.99	97.16	96.30	96.84
	RFE	98.79	98.91	96.77	97.47
	GA	99.16	99.35	97.09	97.69
	DMI-GA	99.29	99.46	97.41	97.90
SVM	NoFS	94.50	94.56	93.68	95.41
	CS	98.80	98.88	96.95	97.32
	CFS	95.44	95.89	94.79	95.13
	IG	98.27	98.45	96.39	97.12
	SU	96.93	97.01	96.22	96.84
	RFE	98.58	98.66	96.52	97.20
	GA	99.00	99.09	97.00	97.41
	DMI-GA	99.22	99.40	97.38	97.61
K-Means	NoFS	93.07	93.22	93.59	93.63
	CS	96.80	96.89	95.99	96.10
	CFS	95.59	95.67	94.66	94.66
	IG	95.83	96.15	95.72	95.83
	SU	95.71	96.07	95.67	95.80
	RFE	96.76	96.85	95.94	96.08
	GA	97.19	97.22	96.04	96.16
	DMI-GA	97.73	97.80	96.27	96.33
RF	NoFS	95.09	95.44	94.05	94.42
	CS	99.41	99.70	97.14	97.70
	CFS	95.77	96.19	95.20	95.41
	IG	98.48	98.69	96.67	97.65
	SU	97.70	97.88	96.52	97.18
	RFE	99.11	99.55	96.99	97.68
	GA	99.55	99.79	97.41	97.90
	DMI-GA	99.61	**99.91**	97.77	**98.10**

FS results in significantly lower accuracy, reinforcing the importance of feature selection in improving the ML model performance.

The use of both CV strategies allowed us to evaluate how sensitive the model is to data partitioning. We observed consistent accuracy and F1-score values across both CV methods, indicating the stability of the proposed feature selection approach, DMI-GA. Furthermore, by averaging the performance metrics across all folds in each method, we mitigated the risk of overfitting and obtained a more reliable estimate of real-world performance.

In this study, we have also investigated the relative performance of seven machine learning models with chosen eight feature selection methods across two cross-validation strategies, using a robust ranking mechanism [101] to evaluate the average percentage gain in the accuracies of each model. The ranking was determined based on the average increase in performance compared to the lowest accuracy (P_L) achieved by all other combinations of methods. For n combinations of classifiers and feature selection methods, including those without feature selection, the average percentage gain in accuracy ($Gain_i$) for combination i is calculated as:

$$Gain_i = \frac{1}{n}\sum_{c=1}^{n}\frac{acc_c - P_L}{P_L} \times 100 \tag{3.10}$$

where, acc_c is the intrusion detection accuracy of the model for the i^{th} combination and ML model c.

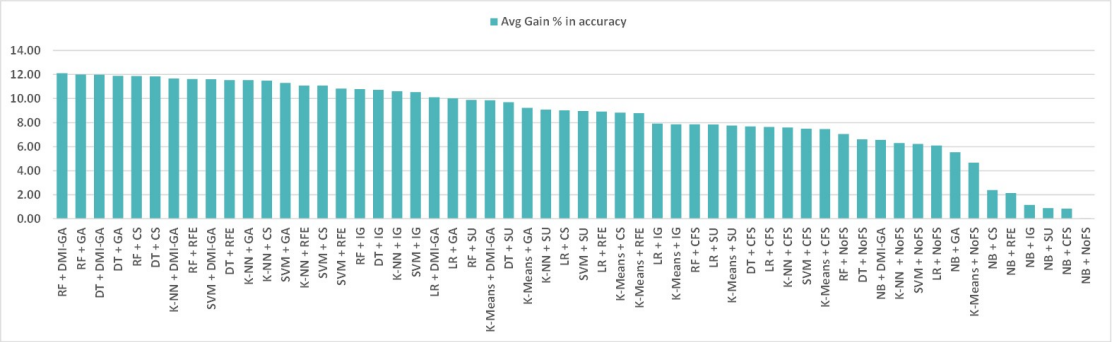

Fig. 3.8: Ranking of the combination of ML and FS methods based on Performance Gain in Accuracy (%)

Fig. 3.8 illustrates the ranking of 56 models using a combination of ML and FS methods, sorted by their relative accuracy gain values in descending order. Notably,

the combination of random forest and the novel DMI-GA method exhibits the highest relative performance gain based on achieved detection accuracy. The analysis further indicates that the performance is consistently superior when using a combination of ML and FS methods compared to scenarios where no feature selection is applied. These results also suggested that evolutionary-based feature selection methods like DMI-GA and GA effectively enhance performance when paired with robust classifiers like RF and DT. It can therefore be concluded that the model's performance improves in detecting intrusions when applied to the optimal feature set obtained through the novel hybrid feature selection method, DMI-GA.

3.3.5 Statistical Significance

To establish the robustness of our experimental results, statistical significance tests were performed on the outcomes of the feature selection techniques when evaluated with individual machine learning models. This is done using two statistical tests, the Friedman test [102] and paired Wilcoxon signed-rank test with Holm correction [103].

The goal is to test whether the proposed DMI-GA feature selection method yields significantly better performance than the other FS methods. To analyse this, we performed the statistical tests separately for each ML model and across the 10 CV folds using accuracy. The tests are performed at a standard significance level, $\alpha = 0.05$ and a confidence interval of 95%.

The non-parametric Friedman test was employed to evaluate whether statistically significant differences existed among the feature selection techniques. It compared the eight feature selection methods across 10 CV folds to test the null hypothesis (H_0) that all feature selection techniques perform equivalently. The alternative hypothesis (H_1) indicates that at least one feature selection method differs significantly. Following the rejection of the null hypothesis, post-hoc analysis is conducted to examine pairwise comparisons between the proposed DMI-GA approach and the baseline feature selection methods. For every fixed ML method, the Friedman rank test is evaluated using average ranking R_j, where within each fold $i = 1, \ldots, N$ and method $j = 1, \ldots, k$, let r_{ij} be the rank. The sum of ranks $R_j = \sum_{i=1}^{N} r_{ij}$ and the average rank, $\bar{R}_j = \frac{1}{N} R_j$ for $N = 10$ folds and $k = 8$ feature selection methods, given by Eqn. 3.11.

$$\chi_F^2 = \frac{12N}{k\,(k+1)} \sum_{j=1}^{k} \bar{R}_j^2 - \; 3N\,(k+1) \tag{3.11}$$

Table 3.9 presents the Friedman rank test statistics, including the F-statistic values (F_r) and corresponding *p*-values, for each machine learning classifier when evaluated with the eight feature selection techniques under 7 degrees of freedom. In all cases, the *p*-values are well below the 0.05 significance level, leading to the rejection of the null hypothesis and confirming that statistically significant performance differences exist among the methods. The relative values of the $F_r = \chi_F^2$ statistics also highlight classifier-specific sensitivity to feature selection. For instance, SVM and Logistic Regression show greater dependence on the choice of features compared to Random Forest, which appears more robust across feature subsets. To summarise, all classifiers experience statistically significant performance improvements due to feature selection, with the strongest effects in SVM, LR, kNN, and NB, while the least in RF, but still show significant improvement. Consequently, a pairwise post-hoc test, the Wilcoxon signed-rank with Holm correction, was performed to identify which feature selection method significantly outperforms the other.

Table 3.9: Friedman Rank Test Results

ML Model	F_r	**p-value**
LR	32.448	0.000762
DT	22.631	0.000901
NB	30.509	0.000119
kNN	32.400	0.000786
SVM	34.176	0.000301
Kmeans	28.457	0.000769
RF	18.933	0.000746

The post-hoc Wilcoxon-Holm correction analysis presented in Table 3.10 demonstrates that the proposed DMI-GA technique gives statistically significant improvements over most baseline FS methods across the majority of ML models, along with the Holm-adjusted *p-values*. Although DMI-GA achieves results statistically comparable to GA and RFE in certain models, it consistently avoids performance degradation across classifiers, unlike other methods, which show variability. This stability highlights DMI-GA

Table 3.10: Post-hoc Wilcoxon-Holm Test Results Comparing DMI-GA with Baseline FS Methods across ML models

FS Comparison	LR		DT		NB		kNN		SVM		KMeans		RF	
	p-value	Reject	p-value	Reject	p-value	Reject	p-value	Reject	p-value	Reject	p-value	Reject	p-value	Reject
DMI-GA vs NoFS	0.0021	Y	0.0024	Y	0.0035	Y	0.0044	Y	0.0009	Y	0.0023	Y	0.0020	Y
DMI-GA vs CFS	0.0013	Y	0.0127	Y	0.0067	Y	0.0091	Y	0.0076	Y	0.0124	Y	0.0160	Y
DMI-GA vs Chi2	0.0047	Y	0.0068	Y	0.0043	Y	0.0845	N	0.0142	Y	0.0087	Y	0.0039	Y
DMI-GA vs IG	0.0065	Y	0.4892	N	0.0032	Y	0.0047	Y	0.0218	Y	0.0046	Y	0.0105	Y
DMI-GA vs SU	0.0036	Y	0.0041	Y	0.0055	Y	0.0033	Y	0.0044	Y	0.0159	Y	0.2324	N
DMI-GA vs RFE	0.0054	Y	0.0364	Y	0.0094	Y	0.0092	Y	0.0305	Y	0.0212	Y	0.0873	N
DMI-GA vs GA	0.0879	N	0.2310	N	0.0126	Y	0.2718	N	0.0439	Y	0.0335	Y	0.1934	N

Y = Yes, N = No.

as not only an effective but also a highly reliable feature selection method.

Table 3.11: Average Ranking of Feature Selection Methods

Feature Selection Method	Average Ranking
DMI-GA	1.2
GA	2.8
CFS	3.3
IG	3.7
SU	5.5
RFE	5.9
CS	6.6
NoFS	8.0

Table 3.11 presents the average Friedman ranking for different feature selection methods across 10 CV folds for each ML method. The lower the rank, the better the feature selection method. The ranking results reveal that the proposed DMI-GA achieved the highest consistency across folds, outranking the others.

3.3.6 Performance Comparison of Proposed Framework with existing IDS models

In addition to comparing the proposed solution with other feature selection models and machine learning techniques, we conducted an experiment to compare the DMI-GA method with recent state-of-the-art methods for intrusion detection, presented in Table 3.12, based on the model's detection accuracy and F1-score. These studies are selected for comparison based on their relevance and recency in the field of intrusion detection, the methodological similarity to our approach, their established performance benchmarks, and the diversity of techniques they encompass. This selection ensures a comprehensive evaluation of the proposed DMI-GA method against state-of-the-art solutions. It can be observed that the proposed intrusion detection model, RF-DMI-GA, achieved the highest accuracy, demonstrating superior performance in detecting network intrusions against the existing intrusion detection models when implemented on the original set of features.

Table 3.12: Performance analysis of the RF-DMI-GA intrusion detection model with existing models on the original dataset

Ref.	Detection Model	FS Method	Features	Accuracy (%)	F1 (%)
Halim et. al. [39]	SVM	GbFS	14	99.83	97.78
Awad and Fraihat [41]	RF	DT-RFECV	12	95.27	93.99
Turukmane et. al. [42]	M-MultiSVM	ONgO	12	99.79	97.63
Akhiat et. al. [43]	RF	IDS-EFS	10	98.88	97.51
Proposed	RF	DMI-GA	**18**	**99.91**	**98.10**

3.3.7 Performance Comparison of Proposed Framework on Various Benchmark Datasets

The proposed framework is also compared with several existing intrusion detection systems, discussed in subsection 2.1.1, on five publicly available datasets, including KDD Cup'99 [104, 105], UNSW-NB15 [106], CIC-IDS2018 [16], IOTID20 [107], and CIC-IoT2023 [108], with performance assessed based on detection accuracy, presented in Table 3.13. These datasets are widely used in IDS research and provide a standardized benchmark for evaluating the effectiveness of different approaches. They differ in network traffic characteristics, attack types, and feature sets, providing a comprehensive foundation for comparative analysis. By evaluating the framework across these diverse datasets, the study highlights the robustness and generalizability of the proposed model in varied network environments.

The IoTID20 dataset includes 83 features with labelled benign and malicious traffic collected from a smart home IoT setup, while the CIC-IoT2023 dataset comprises 47 features and represents diverse, realistic IoT attack scenarios. These datasets were utilized to assess the generalizability and robustness of the proposed RF-DMI-GA model in modern IoT environments, beyond conventional network-level flows, demonstrating its effectiveness against emerging threats in resource-constrained and heterogeneous IoT systems.

In [4], we performed a comparative analysis for intrusion detection using DT, NB, RF, SVM, LR, and KNN classifiers. Binary classification of benign and intrusive data was carried out after applying feature importance-based selection using Gini impor-

Table 3.13: Performance Analysis of the RF-DMI-GA Model with existing IDS models on five benchmark datasets

Ref.	ML Model	FS Method	KDD Cup'99		UNSW-NB15		CSE-CIC-IDS2018		IoTID20		CIC-IoT2023	
			Features selected	Accuracy (%)	Features selected	Accuracy (%)	Features selected	Accuracy (%)	Features selected	Accuracy (%)	Features selected	Accuracy (%)
[31]	SVM	GA	5	96.61	6	95.46	7	94.24	20	88.10	17	97.16
[34]	RF	Chi-square	20	99.91	19	99.64	16	99.90	45	99.17	29	99.69
[41]	RF	DT-RFECV	12	96.89	15	95.30	14	97.11	31	89.99	22	99.30
[42]	M-MultiSVM	ONgO	11	99.07	12	97.53	12	99.89	24	98.99	19	99.70
[43]	RF	IDS-EFS	9	99.00	11	89.91	11	89.87	29	85.79	19	96.44
[45]	XGBoost	Fusion FS	9	94.99	8	92.42	10	95.29	27	89.48	20	98.82
[46]	RF	Gini FI	12	99.84	10	97.16	13	99.77	37	95.22	26	99.55
[109]	RF	RFE	22	99.65	28	98.11	36	99.55	33	94.78	30	99.57
[4]	LR	Gini FI	6	99.04	11	98.89	8	99.25	28	92.06	11	99.31
Proposed	RF	DMI-GA	15	**99.90**	17	**99.86**	15	**99.93**	20	**99.82**	21	**99.87**

tance. A comparative analysis of this model on various benchmark datasets is presented in Table 3.13.

The results emphasize the critical role of feature selection in enhancing intrusion detection accuracy. The proposed RF-DMI-GA model consistently outperforms existing models, achieving the highest accuracy across all the datasets: 99.90% for KDD Cup'99, 99.86% for UNSW-NB15, 99.30% for CSE-CIC-IDS2018, 99.82% for IoTID20, and 99.87% for CIC-IoT2023 dataset. This suggests that the DMI-GA feature selection method effectively enhances classification performance. Additionally, while other models like LR with Gini Feature Importance (FI) and RF with DT-RFECV also exhibit high accuracy, they generally require a higher computational complexity to achieve an optimal number of features, indicating that the proposed method achieves a better feature set efficiently.

3.4 Chapter Summary

This chapter presents a comprehensive analysis of the anomaly-based Network Intrusion Detection Systems (NIDS), emphasizing the impact of feature selection on detection performance. It introduces a novel Dynamic Mutual Information-based Genetic Algorithm (DMI-GA), which integrates mutual information and genetic algorithms to select the most relevant features while minimizing redundancy. The study evaluated the efficiency of DMI-GA against six existing feature selection techniques, including Chi-Squared, Information Gain, Correlation Feature Selection, Recursive Feature Elimination, and Genetic Algorithm, as well as a baseline with no feature selection. The proposed framework consists of three phases: data collection and pre-processing, feature extraction, selection, and ranking, followed by intrusion detection using machine learning models, including Logistic Regression, Decision Tree, Naïve Bayes, k-NN, Random Forest, SVM, and K-Means. Experimental results indicated that DMI-GA consistently outperformed other selection methods, achieving the highest accuracy of 99.91% and F1-score of 98.10% with the Random Forest classifier, which demonstrated its effectiveness in reducing false positives and improving classification precision. Cross-validation using 10-Fold CV and LOOCV further demonstrated the robustness of the approach. Two statistical analysis tests, the Friedman rank test and the post-hoc Wilcoxon-Holm correction test, were performed to validate the results obtained and confirming that the

proposed feature selection method gives dominant results than no feature selection or other baseline methods. A comparative study with recent state-of-the-art intrusion detection models confirmed that RF-DMI-GA surpasses existing methodologies, establishing it as a highly efficient feature selection-based approach for enhanced intrusion detection in high-dimensional datasets. Performance analysis across various benchmark datasets also revealed the robustness of the proposed framework. This chapter focused on the binary intrusion detection to establish a strong baseline. Multiclass categorization is addressed in the following chapter, while future work will extend evaluation to include computational profiling and cross-dataset generalization. Moreover, the next chapter explores the challenge of the high dimensionality of the dataset.

Chapter 4

Enhanced NIDS Leveraging Modified Picture Fuzzy Clustering

This chapter presents an anomaly-based NIDS framework that addresses the curse of high dimensionality in datasets and is evaluated using a robust benchmark dataset encompassing modern attack types. This study proposes a modified Picture Fuzzy Clustering technique, $mP_{ic}FC$ designed to improve the efficiency of intrusion detection. The optimization problem is formulated and solved to determine the prototypes of the clusters. In addition, this chapter provides experimental results, analysis, and statistical validation for attack classification.

4.1 Introduction

Intrusion Detection Systems have proven to be an effective mechanism to monitor and detect potential attacks, as well as abnormal activities, to mitigate the risks associated with network attacks. In the past, several state-of-the-art machine learning-based approaches have been developed to detect network intrusions in order to safeguard one's information. Researchers have been striving to develop intrusion detection techniques that are not only more efficient but also more robust. Furthermore, these conventional models employ either supervised or unsupervised machine learning techniques on a lower-dimensionality dataset and thus are incapable of handling a higher-dimensional data space. Due to the higher dimensionality of the data, the ML models are affected

more severely, leading to a higher false alarm rate (FAR). Most of the traditional intrusion detection methods also fail to detect the types of attacks present in the network or are capable of finding a few of the existing ones [110, 56]. The datasets used in these conventional models are significantly outdated, limiting the efficiency of the IDS, particularly when it comes to detecting rarely encountered or recent attack types [49]. Since dataset quality is essential for accurately classifying and detecting intrusions, these datasets containing both anomalous and benign network data packets are often highly imbalanced, which can reduce classification performance [111, 112].

Higher dimensional data interfere with the performance of the detection model and, as a result, produce a lower detection rate of finding anomalous network traffic. Over the past few years, Fuzzy Clustering has been widely used in the field of intrusion detection [113, 114]. Clustering is an unsupervised ML technique that proves to be a highly effective approach for analyzing data and extracting valuable insights. The fuzzy cluster method can be used for network traffic flow analysis by examining and grouping network traffic flow patterns.

This study proposes an ML–mP$_{ic}$FC model, a novel approach for anomaly-based network intrusion detection that leverages machine learning techniques and picture fuzzy clustering on a dimensionality-reduced dataset. Our approach focuses not only on intrusion detection, but also on reducing the dimensionality of the dataset and detecting the current type of network attacks.

The Picture Fuzzy C-Means clustering method (FC-P_cFS), based on Picture Fuzzy Sets, is proposed [115]. Its performance is highly sensitive to the initialization of key parameters, including initial cluster centers, fuzzifier, membership partition matrix, and neutrality matrix. However, improper initialization can lead to suboptimal results and impact the convergence time of the algorithm.

The novel mP$_{ic}$FC approach introduces an additional layer of decision making that helps to handle uncertainty more efficiently. By distinguishing between partial membership and complete non-membership, the mP$_{ic}$FC allows for more precise classifications. The refusal or hesitation degrees help reduce bias in clusters, ensuring that uncertain data points do not unduly influence the clustering results. Furthermore, the proposed framework addresses the class imbalance problem in the dataset to reduce bias toward the majority class and enhance the model's performance. Also, the robustness of the proposed framework is investigated in contrast to other related methods on the

CSE-CIC-IDS2018 dataset. The key contributions of the chapter are summarized as:

1. We proposed a ML–mP$_{ic}$FC model for anomaly-based network intrusion detection based on Machine Learning (ML) techniques and modified Picture Fuzzy Clustering (mP$_{ic}$FC) method.

2. Binary classification using four well-known ML classifiers was performed on the dimensionality-reduced optimal dataset, which classifies the network attacks from the benign ones.

3. We proposed a novel Picture Fuzzy Clustering technique, mP$_{ic}$FC, to detect and cluster the various types of attacks present in the anomalous network. Experiments were carried out on the up-to-date dataset CSE-CIC-IDS2018, which helped detect the latest network attacks.

4. We utilized the kernel function as a distance measure to compute the distance between the center of the cluster and the data samples, as it aids in reducing the effect of noise.

5. Principal component analysis (PCA) and linear discriminant analysis (LDA) are used to minimize the effect of high dimensionality on the dataset and further improve overall performance by reducing time delay, computational resources, and computational cost.

6. We validated the performance of the ML classifiers using Leave-One-Out Cross-Validation (LOOCV) and k-fold validation techniques, with k values set to 5 and 10, on balanced and imbalanced datasets. Additionally, a statistical analysis was performed using the Friedman test and a post-hoc Wilcoxon-Holm test. We subsequently analysed the performance of the proposed framework in comparison to other related state-of-the-art models on the aforementioned dataset.

This chapter is organized as follows. In section 4.2, various notations and preliminary concepts used in our work are discussed. Section 4.3 describes the proposed modified Picture Fuzzy Clustering method for intrusion detection. Section 4.4 presents the experimental results, performance validation, and statistical analysis. Also, a comparison of the performance evaluation of the proposed model with the existing ones is

presented in this section. The study is summarized in section 4.5, which provides a summary of the chapter.

4.2 Preliminaries and Notations

This section gives a brief description of the prerequisites related to Picture Fuzzy Sets, clustering principles, and key mathematical notations essential for understanding the proposed methodology. It serves as a foundation for the development and application of mP$_{ic}$FC in the anomaly-based network intrusion detection.

- **Fuzzy Sets (FS):** The fuzzy set *S* is defined over the universe-of-discourse *X* and with $\mu_S(x)$ as the membership function which represents the degree of belongingness of an element in a fuzzy set, for an element $x \in X$ [116], represented by:

$$S = \{\langle x, \mu_S(x)\rangle \mid x \in X\} \tag{4.1}$$

  where, $\mu_S(x)$ is the membership function with $\mu_S(x)$: X $\rightarrow$ [0, 1], whereas $\nu_S(x)$ is the non-membership function defined as:

$$\nu_S(x) = 1 - \mu_S(x) \tag{4.2}$$

- **Intuitionistic fuzzy sets (IFS):** An IFS *F*, over set *X* with $x \in X$ [117], is represented by:

$$F = \{\langle x, \mu_F(x), \nu_F(x)\rangle \mid x \in X\} \tag{4.3}$$

  where $\mu_F(x)$ and $\nu_F(x)$ $\forall x \in X$ satisfy the constraints presented in Eqns. (4.4) and (4.5), as follows:

$$\mu_F(x), \nu_F(x) \in [0, 1] \tag{4.4}$$

$$0 \leq \mu_F(x) + \nu_F(x) \leq 1 \tag{4.5}$$

- **Hesitation:** The presence of uncertainty arose from a lack of sufficient knowledge in defining the membership function of *x* in the IFS *F*, which is called the hesitation degree, defined by:

$$\pi_F(x) = 1 - [\mu_F(x) + \nu_F(x)] \tag{4.6}$$

$$\mu_F(x), \nu_F(x) \in [0, 1] \tag{4.7}$$

An IFS is equivalent to a fuzzy set when $\pi_F(x) = 0$.

- **Sugeno and Yager Generating function operators:** Using Sugeno and Yager generating operators, the non-membership or the negation function can be defined. With the help of the Sugeno function [118], the negation function for the set of membership values given for any element *x*, along with the negation parameter δ, can be defined as follows:

$$\nu_F(x) = Neg(\mu_F(x)) = \frac{1 - \mu_F(x)}{1 + \delta\mu_F(x)}, \delta > 0 \tag{4.8}$$

Whereas, using the Yager generating function [119, 120], it is defined as:

$$\nu_F(x) = Neg(\mu_F(x)) = (1 - \mu_F^{\delta}(x))^{1/\delta}, \delta > 0 \tag{4.9}$$

- **Picture Fuzzy Set (PFS):** A PFS, *P* [121], an extension of IFS, defined over set *X* with $x \in X$ is

$$P = \{\langle x, \mu_P(x), \eta_P(x), \gamma_P(x)\rangle \mid x \in X\} \tag{4.10}$$

where $\mu_P(x), \eta_P(x)$, and $\gamma_P(x)$ are the positive, neutral, and negative degrees of each *x*, which satisfies the following two constraints $\forall x \in X$:

$$\mu_P(x), \eta_P(x), \gamma_P(x) \in [0, 1] \tag{4.11}$$

$$0 \leq \mu_P(x) + \eta_P(x) + \gamma_P(x) \leq 1 \tag{4.12}$$

The degree of refusal of *x* can be evaluated as:

$$\xi_P(x) = 1 - (\mu_P(x) + \eta_P(x) + \gamma_P(x)), \forall x \in X \tag{4.13}$$

Similar to the equivalence of IFS and FS, PFS yields to the conventional IFS when $\xi_P(x) = 0$.

- **Fuzzy c-means (FCM):** In 1984, [122] presented the FCM algorithm, which is an improvisation of k-means algorithms. It allocates a membership value to each data point, determined by the distance between the data point and each cluster center [123]. Let $X = \{x_1, \ldots, x_r\}$ have r data points with c centroids in $V = \{v_1, \ldots, v_i, \ldots, v_j\}$ with $U = \{\mu_{ij}\}_{cXr}$ as the partition matrix. Minimizing the objective function J_m results in the Fuzzy C-Means (FCM) algorithm partitioning the dataset X into c clusters. Minimization of J_m is defined by:

$$minJ_m = \sum_{i=1}^{c}\sum_{j=1}^{r} \mu_{ij}^m \|x_j - v_i\|^2 \tag{4.14}$$

$$\text{s.t.} \quad \sum_{i=1}^{c} \mu_{ij} = 1, 1 \leq j \leq r; \tag{4.15}$$

$$\mu_{ij} \geq 0, 1 \leq i \leq c, 1 \leq j \leq r; \tag{4.16}$$

$$\sum_{j=1}^{r} \mu_{ij} > 0, 1 \leq i \leq c \tag{4.17}$$

where *m* is the fuzzifier constant, which manages the resulting partition's fuzziness. For the i^{th} cluster, the membership value of the j^{th} data point is represented by μ_{ij} and ν_i represents the center of the i^{th} cluster. $\|x_j - v_i\|^2$ represents the distance between the centroid of the i^{th} cluster and the j^{th} data point. By the Lagrangian method of undetermined multiplier [122], the solution of Eqn. (4.14) can be obtained.

- **Intuitionistic Fuzzy c-means (IFCM):** IFCM is a modified version of FCM [124, 125] that performs clustering on IFS. The IFS theory uses an additional uncer-

tainty parameter, hesitation degree, to define the membership function, and therefore, the clusters converge to the desired location more than the ones achieved using FCM [126]. In IFCM, a distance measure is applied as a proximity function, which represents the Euclidean distance between the centroid of the i^{th} cluster and the j^{th} data point, thus presenting the objective function as:

$$\min J_m = \sum_{i=1}^{c}\sum_{j=1}^{r} \mu_{ij}^{m} \|x_{j_{\text{IFS}}} - v_{i_{\text{IFS}}}\|^2 \tag{4.18}$$

$$\text{s.t.} \quad \sum_{i=1}^{c} \mu_{ij} = 1, 1 \leq j \leq r; \tag{4.19}$$

$$\mu_{ij} \geq 0, 1 \leq i \leq c, 1 \leq j \leq r; \tag{4.20}$$

$$\sum_{j=1}^{r} \mu_{ij} > 0, 1 \leq i \leq c \tag{4.21}$$

where $Y = \{y_1, \ldots, y_r\}$ are *r* IFSs, each having *n* data points with *c* centroids in $V = \{v_1, \ldots, v_j\}$ and the Euclidean distance for IFS between $x_{j_{\text{IFS}}}$ and $v_{i_{\text{IFS}}}$ is defined by [127] as:

$$\|x_j - v_i\| = \left[\frac{1}{2}\left\{(\mu(x_j) - \mu(v_i))^2 + (\nu(x_j) - \nu(v_i))^2 + (\pi(x_j) - \pi(v_i))^2\right\}\right]^{1/2} \tag{4.22}$$

- **Picture Fuzzy Clustering (PFC):** PFC is a fuzzy clustering method applied to PFS [115], called FC-PFS. It is an improved variant of IFCM with the optimized objective function defined as follows:

$$\min J_m = \sum_{k=1}^{r}\sum_{j=1}^{c}(\mu_{kj}(2 - \xi_{kj}))^m \ \|X_k - V_j\|^2 + \sum_{k=1}^{r}\sum_{j=1}^{c}\eta_{kj} \ (log \ \eta_{kj} + \xi_{kj}) \tag{4.23}$$

$$\text{s.t.} \quad \sum_{j=1}^{c}(\mu_{kj}(2-\xi_{kj})) = 1; \tag{4.24}$$

$$0 \leq \mu_{kj}, \eta_{kj}, \xi_{kj} \leq 1; \tag{4.25}$$

$$\mu_{kj} + \eta_{kj} + \xi_{kj} \leq 1; \tag{4.26}$$

$$\sum_{j=1}^{c}(\eta_{kj} + \frac{\xi_{kj}}{c}) = 1 \tag{4.27}$$

Where, the first constraint is the new membership-like function that satisfies the sum-row constraint of the traditional FCM algorithm. The last constraint suggests that the model is guaranteed to work as one of the two uncertainties, refusal or neutral degree, exists in the model.

If $\xi_{kj} = 0$ and the last constraint, Eqn. (4.27) is non-existent, then the FC-PFS model generalizes to the conventional IFCM model. When this condition, along with $\eta_{kj} = 0$, is met, the FC-PFS model is generalized to the traditional FCM model.

As the above-mentioned model is employed on PFS, it provides better quality clustering than FS and IFS, and it also performs well for complex real-world structures. Thus, for efficient identification of various types of network attacks, we propose a novel fuzzy clustering model based on the FC-PFS model.

4.3 Proposed Framework

This section describes the proposed framework for an efficient anomaly-based network intrusion detection of the latest network attacks using supervised machine learning models to classify the benign network traffic from the attack classes, and further employed unsupervised machine learning techniques to detect and cluster various network attacks. The study also proposed a novel fuzzy clustering approach, $mP_{ic}FC$ based on the picture fuzzy clustering approach with an aim to create a proficient model that can detect anomalous intrusions in the network.

The framework of the proposed intrusion detection system, developed in three phases involving data pre-processing, binary classification of network traffic, and clustering network attacks, is illustrated in Fig. 4.1. Initially, the captured network traffic was preprocessed, the class imbalance problem was addressed, and, due to the high dimensionality of the dataset, two widely used dimensionality reduction techniques, PCA and LDA, were employed. Further, in the second phase, we investigated the performance of the four widely popular binary classifiers-random forest, SVM, k-NN, and decision tree-on the dimensionality-reduced dataset to classify the network attacks from the benign (or normal) traffic. To validate the performance of these classifiers, we used three Cross-Validation (CV) strategies: 5-fold CV, 10-fold CV, and LOOCV. Once the network attacks were identified, the proposed mP$_{ic}$FC method was used to cluster the various types of attacks in the final phase. These three phases are explained in detail in the following subsections.

4.3.1 Phase I: Data Pre-processing and Dimensionality Reduction

In this work, the experiments were performed on the publicly available network dataset CSE-CIC-IDS2018, which is an up-to-date benchmark and realistic dataset for cyber defence. It was developed collaboratively by the Canadian Institute for Cybersecurity (CIC) and the Communications Security Establishment (CSE). It captures real-world network traffic from both benign and malicious activities, making it well-suited for evaluating anomaly-based intrusion detection models. Network flows in the dataset are captured using CICFlowMeter, which extracts 80 statistical features per flow and each flow is labelled with its traffic class. It was prepared in an infrastructure comprising 50 attacking machines and a victim organization with 420 systems and 30 servers in a time span of 10 days, resulting in capturing 1,62,33,002 network traffic instances. A snippet of the traffic flow in the CSE-CIC-IDS2018 dataset is presented in Fig. 4.2, and the distribution of the dataset consisting of various types of network traffic categories and attack types present in CSE-CIC-IDS2018, along with the network traffic distribution (%) [128] is presented in Fig. 4.3, with approximately 83% of benign network instances and 17% of attack instances. This shows that the dataset is highly imbalanced, which means the classifier is biased towards the majority class.

The dataset, originally in PCAP format, underwent further pre-processing as de-

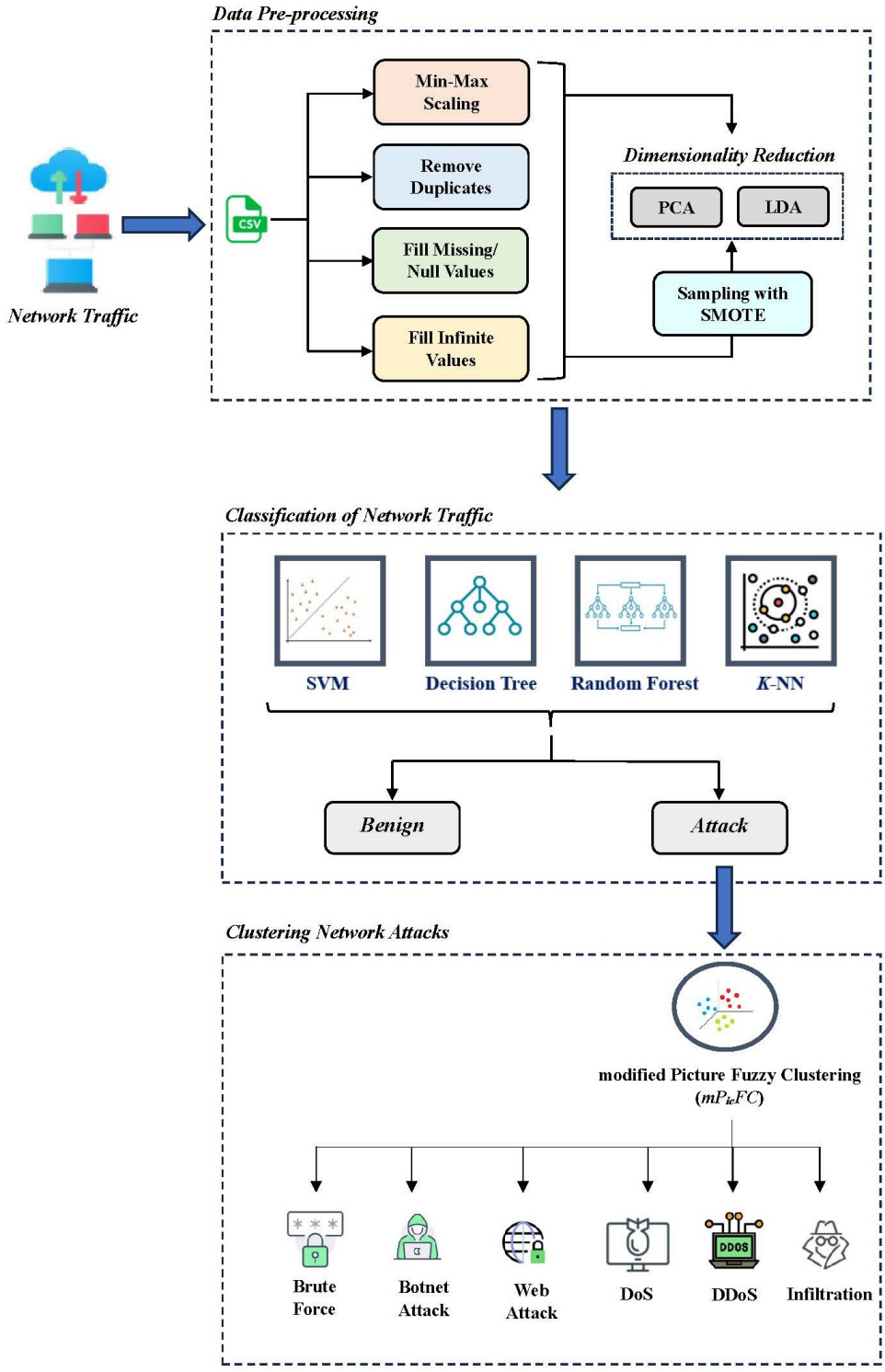

Fig. 4.1: Overall framework of the proposed intrusion detection model

Index	Subflow Bwd Byts	Init Fwd Win Byts	Init Bwd Win Byts	Fwd Act Data Pkts	Fwd Seg Size Min	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label
479	14163	8192	124	7	20	21401	23704.71875	69788	11676	10100000	149619.4334	10200000	9800161	Benign
480	6530	8192	125	7	20	20912.5	22415.06092	66667	11701	10100000	52133.17988	10200000	10100000	Benign
481	6530	8192	125	7	20	21297.83333	23369.21172	69000	11713	10100000	52233.26233	10200000	10100000	Benign
482	5626	8192	119	8	20	16942.16667	12929.81218	43335	11632	10000000	218215.8	10200000	9627055	Benign
483	5602	8192	114	8	20	17032.5	13000.67561	43570	11705	10000000	284122.1587	10200000	9439561	Benign
484	420	8192	51100	8	20	47788.66667	88439.50917	228315	11584	10100000	118975.2199	10200000	9859147	Benign
485	1964	8192	946	9	20	35263.125	11195.85127	62971	31246	10100000	61851.15447	10200000	10000000	Benign

Fig. 4.2: Snippet of network traffic flow in CSE-CIC-IDS2018

tailed in subsection 3.2.1. After cleaning the data, we normalized the dataset using the Min-Max normalization technique [129] and then reduced its dimensionality using two techniques, PCA and LDA. Additionally, experiments were conducted on a dataset that, after cleaning, was class-balanced using the Synthetic Minority Over-sampling Technique (SMOTE) and subsequently underwent dimensionality reduction, described later in this subsection. Throughout, we assume that after cleaning, Min-Max scaling and SMOTE-based balancing, when employed, preserve class-relevant structure prior to PCA or LDA.

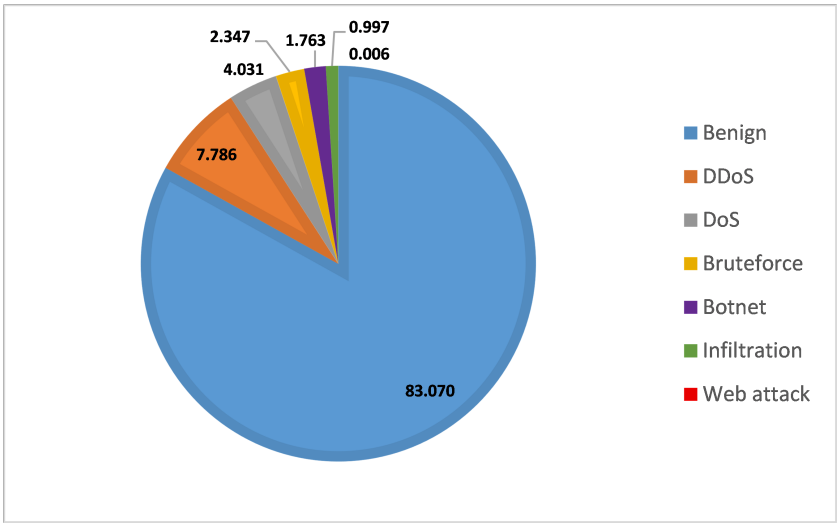

Fig. 4.3: Approximate traffic distribution (%) of CSE-CIC-IDS2018

- **Normalization:** Normalization is a data scaling method of transforming the features in a dataset to a standardized scale. This method helps enhance the efficiency and accuracy of machine learning classifiers, especially in the case of large datasets, since it gives equal weightage to all features. Some of the data normalization techniques include Z-score normalization, Sigmoidal normalization,

Min-Max normalization, Softmax normalization, Max normalization, etc. In this work, we have employed the Min-Max normalization technique, which maintains the exact relationships among data values without introducing any latent bias. Additionally, this technique takes less time and is less complex than the others [130, 131, 132]. Based on this technique, raw data features were scaled to the range of [0, 1]. The normalized value ($X_{normalized}$) is obtained by subtracting the feature's minimum value $X(min)$ from each data point (X) and then dividing the result by the feature's range, as presented below:.

$$X_{normalized} = \frac{X - X(min)}{X(max) - X(min)} \tag{4.28}$$

- **Handling Class Imbalance:** Class imbalance is a challenging issue that occurs when one class in a dataset significantly outnumbers the others, often leading to biased model performance [49]. Handling class imbalance is crucial in intrusion detection as the majority of network traffic is benign, while actual intrusions are rare, creating an uneven distribution of classes. This imbalance can lead models to favour the majority class, resulting in missed detections of intrusions. Various techniques exist to address class imbalance issues, including random oversampling, random undersampling, Synthetic Minority Oversampling Technique (SMOTE), and Adaptive Synthetic Sampling (ADASYN). In this study, we have handled this imbalance problem with the widely-used method, SMOTE. It helps mitigate overfitting in minority classes as, unlike random oversampling, which simply duplicates samples, SMOTE creates synthetic samples by interpolating between existing minority class instances, making the dataset more varied and reducing the risk of overfitting in duplicate samples [111]. It also helped classifiers better recognize minority attacks that are otherwise underrepresented in the dataset, such as infiltration or web attacks. Since we perform dimensionality reduction with PCA and LDA, SMOTE works well with both of them. As PCA captures maximum variance in the data, the presence of additional minority samples from SMOTE helps PCA retain a more balanced representation. LDA is supervised and works to maximize class separability. Synthetic SMOTE samples support this by making the minority class more distinct, which can improve LDA's effectiveness, especially in binary classification.

- **Dimensionality Reduction:** In this work, dimensionality reduction using the most popular methods—PCA [56, 133, 134, 135] and LDA [136, 137]—was applied to the pre-processed dataset to transform it into a lower-dimensional space while preserving the essence of the original data. This was done to reduce computation time while simultaneously improving the model's performance.

  The unsupervised dimensionality reduction technique, PCA, was performed by transforming the original features or variables into newer ones called principal components, while using the principal components of the dataset instead of all existing original variables, ML algorithms converge in a much faster manner; hence reducing the training time of the algorithm. Its goal is to maximize the variance in the dataset captured by each component. Due to such advantageous characteristics, we opt for PCA for dimensionality reduction in this work, as it is a prevailing analysis tool. In the case of LDA, which is a supervised dimensionality reduction technique, it aims to maximize the separability between different classes. It finds a linear combination of features that best separates the classes. For selecting the number of dimensions, PCA focuses on maximizing variance, which is a measure of the spread of data points, while LDA focuses on maximizing class separability, which is a measure of how well classes can be distinguished. PCA can select up to f dimensions, while LDA can select up to $min(C-1, f)$ dimensions for C number of classes and f features.

4.3.2 Phase II: Binary Classification of Network Traffic

Binary classification in network intrusion detection provides higher accuracy, faster detection, and reduced computational costs. It enhances cybersecurity defences by allowing efficient, real-time attack identification while minimizing complexity and false alarms.

In this study, after the dataset was pre-processed, in the second phase, four well-known supervised ML classification models were used to categorize the network traffic as benign or attack. In this work, we compared and analyzed the performance of these binary classifiers, namely, SVM, Decision Tree, Random Forest, and K-Nearest Neighbour, to detect intrusive traffic from the normal. However, for advanced threat intelligence, further categorization can be performed to identify and analyze different types

of attacks. Therefore, in the final phase, the traffic in the network classified as attacks was used to identify and cluster them into different attack classes.

4.3.3 Phase III: Clustering Network Attacks

This subsection provides an in-depth discussion of the proposed clustering approach, mP$_{ic}$FC, which is built upon picture fuzzy clustering principles. This approach is specifically designed to enhance the detection and classification of various network attack types by leveraging the attack dataset obtained from the previous phase. By incorporating picture fuzzy clustering, the method aims to improve the accuracy and reliability of intrusion detection by effectively handling uncertainty and overlapping data points in the dataset. The results of the proposed clustering approach are expected to contribute to the overall efficiency of network security systems by providing a structured categorization of potential threats.

Clustering is an unsupervised ML and data mining technique that helps categorize unlabelled data items into distinct groups called clusters, with each cluster comprising similar data items. It does so with the help of various features of this unlabelled data, drawing a similar pattern and grouping them according to the presence of these similar patterns. Clustering techniques can be categorized as hard and soft clustering techniques [138, 139]. In hard clustering, every data item either belongs to a particular cluster completely or does not. Meanwhile, in soft clustering, data items can belong to multiple clusters depending on the membership value.

One of the commonly used hard clustering techniques is K-means clustering [28, 140], which divides the data points into k clusters according to the chosen distance metric, which is calculated between the data points and the cluster centroids. The data point nearest to the centroid of the cluster is allocated to that specific cluster. Based on the minimum distance, each data point is always assigned to one of the clusters, which is particularly effective when dealing with highly organized data but not real-world data [141].

In fuzzy clustering, a membership value is assigned to every data point for each of the clusters that it belongs to. The membership value lies within [0, 1], which depicts the degree to which the data point represents a cluster. Therefore, fuzzy clustering offers a versatile and robust approach for dealing with real-world data that possesses

vagueness and uncertainty. Soft or fuzzy clustering consists of techniques like fuzzy c-means (FCM) [122], intuitionistic fuzzy c-means (IFCM) [124], etc., as explained in the previous section. Thus, the difference between hard k-means and soft fuzzy c-means, along with its variants, is that k-means clustering clusters the entire set of data points into k clusters, and each data point belongs to a single cluster. In contrast, fuzzy c-means generates *k* clusters and subsequently assigns each data point to these clusters. Still, it also incorporates a factor that quantifies the degree of membership or the strength of affiliation of the data point to each cluster.

4.3.3.1 The modified Picture Fuzzy Clustering method (*mP$_{ic}$FC*)

Picture fuzzy clustering is an improved version of IFCM based on picture fuzzy sets that consider the degree of positive membership, neutral membership, and negative membership, along with the degree of refusal membership. Thus, according to [115] PFSs have better-quality clustering than FS and IFS. Therefore, inspired by the optimal and superior performance of PFC over FCM and IFCM, we employed the mP$_{ic}$FC technique to detect and cluster various types of network attacks. The additional membership degree enhances clustering quality and enables better handling of ambiguous data points. Furthermore, the integration of the Sugeno function for computing the negation membership refines membership calculations, leading to more reliable clustering outcomes.

Even though there are numerous advantages of PFC, for finding the distance between the data points and the cluster centers, it uses Euclidean distance, and experiments have concluded that for clustering. But, Euclidean distances can exhibit higher error rates and greater sensitivity to noise. In order to overcome this limitation, we have chosen a Hypertangent kernel function [142], which has the potential to boost the expressive capacity of linear machines. Its incorporation into clustering algorithms can enhance their clustering capabilities, leading to improved accuracy and effectiveness. This enhancement is particularly beneficial in cybersecurity applications, where network traffic data can be highly complex. Also, we opted for this kernel due to its robust nature towards noise and non-linear structures present in the underlying data.

Overall, mP$_{ic}$FC's ability to handle uncertainty, its robustness against noise, and its effectiveness in dealing with non-linear data structures make it a superior choice for clustering tasks.

As the network data was classified into intrusive and non-intrusive (benign) datasets using ML classifiers, the intrusive data was further clustered into various types of network attacks using the mP$_{ic}$FC method, $\forall i\ 1 \leq i \leq C,\ \forall j\ 1 \leq j \leq N$, given by:

$$\min J_m = \sum_{i=1}^{C}\sum_{j=1}^{N}\left[(\mu_{ij}(2-\xi_{ij}))^m \|\phi(x_j)-\phi(v_i)\|^2\right] + \sum_{i=1}^{C}\sum_{j=1}^{N}\left(\xi_{ij}\eta_{ij}^2 + \eta_{ij}\xi_{ij}^2\right) \tag{4.29}$$

$$\text{s.t.} \qquad 0 \leq \mu_{ij}, \eta_{ij}, \xi_{ij} \leq 1; \tag{4.30}$$

$$0 \leq \mu_{ij} + \eta_{ij} + \xi_{ij} \leq 1; \tag{4.31}$$

$$\sum_{i=1}^{C}(\mu_{ij}(2-\xi_{ij})) = 1; \tag{4.32}$$

$$\sum_{i=1}^{C}(\eta_{ij} + \frac{\xi_{ij}}{C}) = 1 \tag{4.33}$$

where $\|\phi(x_j)-\phi(v_i)\|^2$ is the distance between the kernel spaces $\phi(x_j)$ and $\phi(v_i)$ presented in Eqn. (4.34), and *m* is the fuzzifier constant, and the value of the negation function ξ_{ij} is evaluated using the Sugeno function [118] using Eqn. (4.35).

$$\|\phi(x_j)-\phi(v_i)\|^2 = \phi(x_j)^T\phi(x_j) + \phi(v_i)^T\phi(v_i) - 2\phi(x_j)\phi(v_i) \tag{4.34}$$

$$\xi_{ij} = 1 - \left[\mu_{ij} + \eta_{ij} + \frac{1-(\mu_{ij}+\eta_{ij})}{1+\delta(\mu_{ij}+\eta_{ij})}\right] \tag{4.35}$$

The optimization problem, presented in Eqn. 4.29 is composed of two terms. The first term measures the weighted intra-cluster similarity in kernel space, thus encouraging the formation of compact and coherent clusters, where points closely align with their respective cluster centers. The second term acts as a regularization function that penalizes high values of both non-membership and refusal simultaneously. Its objective is to control the balance between uncertainty and non-affiliation, thereby promoting crisper and more interpretable clustering results. By discouraging ambiguous cluster assignments, this term supports a clearer separation of data into distinct clusters while still respecting the fuzziness inherent to network traffic data.

Derivation of Cluster Prototype:

The Lagrangian method of undetermined multiplier was used to find the optimal solution of the above-stated model. The Lagrangian function $L(\mu_{ij}, \eta_{ij}, v_i, \alpha_j, \beta_j)$ is represented as:

$$\begin{aligned} L = &\sum_{i=1}^{C}\sum_{j=1}^{N}\left[(\mu_{ij}(2-\xi_{ij}))^m \, \|\phi(x_j)-\phi(v_i)\|^2\right] + \sum_{i=1}^{C}\sum_{j=1}^{N}(\xi_{ij}{\eta_{ij}}^2 + \eta_{ij}{\xi_{ij}}^2) \\ &+ \sum_{j=1}^{N}\alpha_j\left(1-\sum_{i=1}^{C}\left(\eta_{ij}+\frac{\xi_{ij}}{C}\right)\right) + \sum_{j=1}^{N}\beta_j\left(1-\sum_{i=1}^{C}(\mu_{ij}(2-\xi_{ij}))\right) \end{aligned} \tag{4.36}$$

where α_j and β_j are the two Lagrange's multipliers for $j = \{1, 2, \ldots, N\}$.

To find the value of μ_{ij}, taking the partial derivative of *L* w.r.t. μ_{ij} and equating it to 0, we have:

$$\frac{\partial L}{\partial \mu_{ij}} = m\,(\mu_{ij})^{m-1}\,(2-\xi_{ij})^m\,\|\phi(x_j)-\phi(v_i)\|^2 - \beta_j(2-\xi_{ij}) = 0 \tag{4.37}$$

$$(\mu_{ij})^{m-1} = \frac{\beta_j}{m}\,\frac{1}{(2-\xi_{ij})^{m-1}}\,\frac{1}{\|\phi(x_j)-\phi(v_i)\|^2} \tag{4.38}$$

$$\mu_{ij} = \frac{1}{(2-\xi_{ij})}\left(\frac{\beta_j}{m}\right)^{1/(m-1)}\left(\frac{1}{\|\phi(x_j)-\phi(v_i)\|^2}\right)^{1/(m-1)} \tag{4.39}$$

$$\mu_{ij}(2-\xi_{ij}) = \left(\frac{\beta_j}{m}\right)^{1/(m-1)}\left(\frac{1}{\|\phi(x_j)-\phi(v_i)\|^2}\right)^{1/(m-1)} \tag{4.40}$$

Taking summation $\sum_{i=1}^{C}$ on both sides, we rewrite the equation as follows:

$$\sum_{i=1}^{C}\mu_{ij}(2-\xi_{ij}) = \left(\frac{\beta_j}{m}\right)^{1/(m-1)}\sum_{i=1}^{C}\left(\frac{1}{\|\phi(x_j)-\phi(v_i)\|^2}\right)^{1/(m-1)} = 1 \tag{4.41}$$

$$\Rightarrow \left(\frac{\beta_j}{m}\right)^{1/(m-1)} = \sum_{i=1}^{C} \left(\frac{1}{\|\phi(x_j) - \phi(v_i)\|^2}\right)^{-1/(m-1)} \frac{1}{(2-\xi_{ij})} \tag{4.42}$$

Substituting the value of $\left(\frac{\beta_j}{m}\right)^{1/(m-1)}$ in Eq. (4.40) and further solving for μ_{ij}, we get:

$$\mu_{ij} = \frac{1}{(2-\xi_{ij})} \frac{\left(\frac{1}{\|\phi(x_j)-\phi(v_i)\|^2}\right)^{1/(m-1)}}{\sum_{i=1}^{C}\left(\frac{1}{\|\phi(x_j)-\phi(v_i)\|^2}\right)^{1/(m-1)}} \tag{4.43}$$

Similarly, taking the partial derivative of *L* from Eqn. (4.36) w.r.t. η_{ij} and equating it to 0 to find the value of η_{ij}:

$$\frac{\partial L}{\partial \eta_{ij}} = 0 + 2\,\eta_{ij}\,\xi_{ij} + {\xi_{ij}}^2 - \alpha_j = 0 \tag{4.44}$$

$$\alpha_j = 2\,\eta_{ij}\,\xi_{ij} + {\xi_{ij}}^2 \tag{4.45}$$

Also, let $\frac{\partial L}{\partial \xi_{ij}} = 0$, we get:

$$\frac{\partial L}{\partial \xi_{ij}} = m\,{\mu_{ij}}^m \|\phi(x_j) - \phi(v_i)\|^2\,(2-\xi_{ij})^{m-1}(-1) + {\eta_{ij}}^2 + 2\eta_{ij}\xi_{ij} - \frac{\alpha_j}{C} + \beta_j\,\mu_{ij} = 0 \tag{4.46}$$

$$\frac{\partial L}{\partial \xi_{ij}} = \mu_{ij}\left[-m\,{\mu_{ij}}^{m-1}\,(2-\xi_{ij})^{m-1}\,\|\phi(x_j) - \phi(v_i)\|^2 + \beta_j\right] + {\eta_{ij}}^2 + 2\,\eta_{ij}\,\xi_{ij} - \frac{\alpha_j}{C} = 0 \tag{4.47}$$

From Eq. (4.37), we get:

$${\eta_{ij}}^2 + 2\,\eta_{ij}\,\xi_{ij} - \frac{\alpha_j}{C} = 0 \tag{4.48}$$

Therefore, from Eqns. (4.44) to (4.48), we obtain the value of η_{ij} as:

$$\eta_{ij} = \frac{2\eta_{ij}\xi_{ij} + {\xi_{ij}}^2}{C[\eta_{ij} + 2\xi_{ij}]} \tag{4.49}$$

To derive the value of the cluster centers v_i, we apply the kernel and rewrite the Eqn. (4.36) as:

$$L = \sum_{i=1}^{C}\sum_{j=1}^{N}(\mu_{ij}(2-\xi_{ij}))^m \left(1 - tanh\left(\frac{-\|x_j - v_i\|^2}{\sigma^2}\right)\right) + \sum_{i=1}^{C}\sum_{j=1}^{N}(\xi_{ij}{\eta_{ij}}^2 + \eta_{ij}{\xi_{ij}}^2) + \sum_{j=1}^{N}\alpha_j\left(1 - \sum_{i=1}^{C}\left(\eta_{ij} + \frac{\xi_{ij}}{C}\right)\right) + \sum_{j=1}^{N}\beta_j\left(1 - \sum_{i=1}^{C}(\mu_{ij}(2-\xi_{ij}))\right) \tag{4.50}$$

Equating the partial derivative of Eqn. (4.50) to 0, i.e. $\frac{\partial L}{\partial v_i} = 0$, we get:

$$\sum_{j=1}^{N} 2(\mu_{ij}(2-\xi_{ij}))^m \left(1 - tanh\left(\frac{-\|x_j - v_i\|^2}{\sigma^2}\right)\right)\left(\frac{\|x_j - v_i\|}{\sigma^2}\right) = 0 \tag{4.51}$$

Subsequently solving Eqn. (4.51), we get:

$$v_i = \frac{\sum_{j=1}^{N}(\mu_{ij}(2-\xi_{ij}))^m \left(1 + tanh\left(\frac{-\|x_j - v_i\|^2}{\sigma^2}\right)\right) K(x_j, v_i)x_j}{\sum_{j=1}^{N}(\mu_{ij}(2-\xi_{ij}))^m \left(1 + tanh\left(\frac{-\|x_j - v_i\|^2}{\sigma^2}\right)\right) K(x_j, v_i)} \tag{4.52}$$

where $K(x_j, v_i) = \phi(x_j)^T\phi(v_i)$ is the Hypertangent Kernel function, using this, Eq. (4.34) can be rewritten as:

$$\|\phi(x_j) - \phi(v_i)\|^2 = K(x_j, x_j) + K(v_i, v_i) - 2K(x_j, v_i) \tag{4.53}$$

Also, the parameter σ^2 is the dataset's degree of separation and is computed as $\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\|z_i - \bar{z}\|^2$, where $\bar{z} = \frac{1}{N}\sum_{k=1}^{N} z_k$.

The above-proposed methodology and various stages involved in finding the solution of the proposed mP$_{ic}$FC, as discussed in this section, are outlined in Algorithm 4.1.

Algorithm 4.1 Proposed mP$_{ic}$FC method

Input: Network data with *N* data points for C clusters; fuzzifier constant m; max iterations $maxSteps$; threshold ϵ.
Output: μ, η, ξ, and cluster centers v_i.
procedure MP$_{ic}$FC(Novel modified Picture Fuzzy Clustering)
    $k \leftarrow 1$
    Initialize cluster centers ${v_i}^k \leftarrow random$
    repeat
        for $k \leftarrow k + 1$ **do**
            Compute $\mu_{ij}^{(k)}$ membership values for each data point w.r.t. centroids by Eqn. (4.43), $1 \leq i \leq C$, $1 \leq j \leq N$.
            Compute ${\eta_{ij}}^{(k)}$ membership values for each data point w.r.t. centroids by Eqn. (4.49), $1 \leq i \leq C$, $1 \leq j \leq N$.
            Compute ${\xi_{ij}}^{(k)}$ membership values for each data point w.r.t. centroids by Eqn. (4.35), $1 \leq i \leq C$, $1 \leq j \leq N$.
            Compute the cluster centroids ${v_i}^{(k)}$, $1 \leq i \leq C$ by Eqn. (4.52).
        end for
    until $\|\mu^{(k)} - \mu^{(k-1)}\| + \|\eta^{(k)} - \eta^{(k-1)}\| + \|\xi^{(k)} - \xi^{(k-1)}\| < \epsilon$ or $i \geq maxSteps$
end procedure

4.4 Experimental Results and Analysis

In this section, we present a detailed discussion and comparative analysis of the proposed framework against state-of-the-art methods using various evaluation metrics. The performance of four ML classifiers was then compared on both the original and class-balanced datasets. Lastly, we analyzed the performance of widely used clustering techniques, including K-Means, FCM, IFCM, and PFC, in comparison with the proposed mP$_{ic}$FC model. To evaluate the efficiency of our approach, we conducted experiments on the publicly available, up-to-date network traffic dataset CSE-CIC-IDS2018.

For the implementation of the proposed intrusion detection framework, experiments were carried out on a 64-bit Windows 10 operating system, i7 processor with 16 GB RAM. 64-bit Spyder was used for conducting the experiments, with Python v3.7 as the programming language for the implementation of the model. A packet sniffer, Wireshark, was used to capture network data and convert pcap files to the desired CSV format.

4.4.1 Performance Evaluation Metrics

The performance of the proposed method is assessed with the help of various evaluation metrics like accuracy, precision, F1 score, false positive rate (FPR), and false negative rate (FNR), along with three cluster validity indices.

Cluster validity indices are the measures used for the validation of the quality of the clustering algorithms [143, 144]. Since mP$_{ic}$FC extends fuzzy clustering with additional membership degrees, evaluating clustering compactness and separation is crucial. These cluster validity indices help assess how well-defined and meaningful the generated clusters are.

The three cluster validity indices that we have used for clustering are partition coefficient (PC), partition entropy (PE), and Xie-Beni function (XB). PC and PE, cluster validity indices are evaluated using membership values. PC value lies between $\left[\frac{1}{C}, 1\right]$ where *C* is the number of clusters, whereas the value of PE varies from $[0, \log_a C]$. The higher the value of PC or the lower the value of PE, the more optimal clusters are accomplished [145, 146]. XB index [147] is based on compactness and separability, was later modified by [148] with the widely used fuzzifier constant value as $m = 2$ and is represented in Eqn. (4.54). The larger the value of PC, the smaller the degree of overlap amongst the clusters, which indicates that the clusters are well separated. Meanwhile, a lower value of PE indicates a lesser degree of fuzziness in the clusters, tending to a more well-defined cluster. The numerator of XB represents the fuzzy partition's compactness, and the denominator depicts the separation strength amongst the clusters. The lower the XB value, the more optimal the cluster is. PC, PE, and XB are defined and evaluated as:

$$PC = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{N}{\mu_{ij}}^2 \tag{4.54}$$

$$PE = -\frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{N}\mu_{ij}\, log(\mu_{ij}) \tag{4.55}$$

$$XB = \frac{\sum_{i=1}^{C}\sum_{j=1}^{N}{\mu_{ij}}^m \|x_j - v_i\|^2}{n(min_{i,j=1,2,..,C,i\neq j}\|v_i - v_j\|^2)} \tag{4.56}$$

To statistically validate and compare the performance of the four binary classification models, we used three cross-validation methods, k–fold CV ($k = 5$ and $k = 10$), and LOOCV. The primary objective of these methods is to prevent overfitting, which offers an additional robust estimate of the performance of the model. It utilizes all data points and helps reduce biases, making it a more data-efficient method.

Table 4.1 describes all the notations and symbols used in this work.

4.4.2 Binary Classification Performance on Dimensionality-Reduced Datasets

Two experiments were conducted in this study to classify the network data using four ML techniques on both the original and balanced versions of the CSE-CIC-IDS2018 dataset. In the first experiment, the classification was performed on the original and balanced datasets without dimensionality reduction. In the second experiment, classification was carried out on the datasets after dimensionality reduction.

We adopted the same hyperparameter protocol as in the previous chapter 3.3.1. Similar search spaces that produced stable plateaus were used, and then a narrow confirmatory grid was performed on the current approach. Fig. 4.4 demonstrates confirmatory hyperparameter heatmaps with a 10-fold validation mean. The SVM displays a clear ridge for $C \in [10, 14]$ with $tol = 10^{-3}$, supporting $C = 12$ and $tol = 10^{-3}$. For RF, we observe that accuracy improves steadily as the number of trees increases from 50 to around 200, after which the gains begin to saturate, indicating diminishing returns. The chosen default of 100 trees already falls within the stable region, providing reliable accuracy without unnecessary computation. The Decision Tree improves with depth approximately up to 20 and then stabilizes or slightly dips, and smaller split thresholds perform better. kNN supports the Minkowski metric and shows a shallow optimum for $k \in 5, 7$. Table 4.2 presents the various parameters and their corresponding values for the ML classifiers chosen in the experiments of the proposed framework, ensuring optimized performance for classification tasks.

Table 4.1: Notations used in the proposed work.

Notation	Description	Notation	Description
S	Fuzzy Set	X	Universe-of-discourse
$\mu_S(x)$	Membership function of the fuzzy set *S*, which assigns a degree of belongingness to the element *x*	$\nu_S(x)$	Non-membership function or negation function of the fuzzy set *S*
x	An element of Universe-of-discourse X i.e x $\in X$.	ϵ	Threshold Value
F	Intuitionistic fuzzy set (IFS)	P	Picture Fuzzy Set (PFS)
$\pi_F(x)$	Hesitation degree associated with an element *x* in the intuitionistic fuzzy set *F*	$Neg(\mu_F(x))$	Negation function applied to the membership function $\mu_F(x)$, which transforms the membership degree into a non-membership degree
δ	Negation parameter	r	Number of data points in *X*
$\mu_P(x)$	Positive membership degree of *x*, $x \in P$	$\eta_P(x)$	Neutral membership degree of *x*
$\gamma_P(x)$	Negative membership degree of *x*	$\xi_P(x)$	Degree of refusal
C	Number of clusters	V	Set of cluster centroids with v_i as the center of i^{th} cluster, i = 1,2,. . . ,C
μ_{ij}	Membership value of data point i^{th} data point of the j^{th} cluster	U	Partition matrix which contains membership values $_{ij}$, $i = 1, 2, \ldots, c$, $j = 1, 2, \ldots, r$
J_m	Objective Function	m	Fuzzifier constant
Y	Set of *r* IFSs	p	Number of dimensions in the dataset
ϕ_{xj}	Mapping of data point x_j in kernel space	$K(x_j, v_i)$	Hypertangent Kernel function
α_j, β_j	Lagrange multipliers, $j = 1, 2, 3..., N$	σ^2	Dataset's degree of separation
$\bar{z}$	Mean of all data points in the dataset	$maxSteps$	Maximum number of iterations
$\tanh(\cdot)$	Hyperbolic tangent function used in the Hypertangent kernel function		

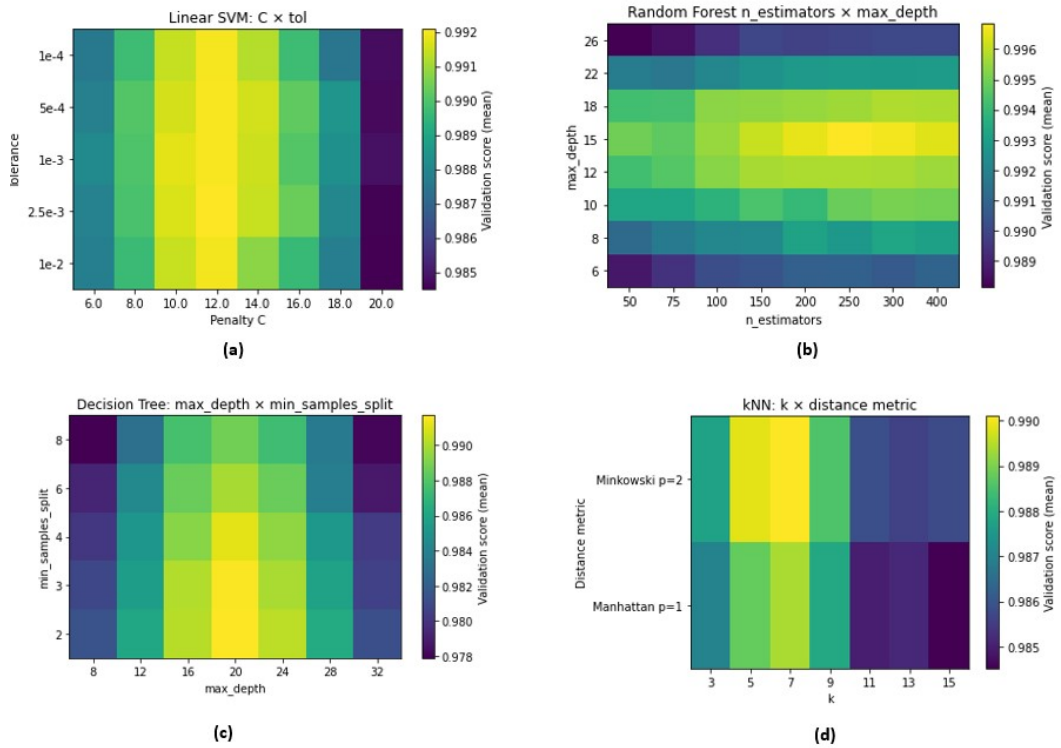

Fig. 4.4: Hyperparameter tuning heatmaps for ML Classifiers, SVM, RF, DT, and kNN.

The penalty parameter for SVM is set to 12, striking a balance between misclassification tolerance and model complexity, while the linear kernel is chosen for its efficiency in handling linearly separable data. A stopping tolerance of 0.001 ensures convergence without unnecessary iterations, and the maximum iteration limit of 1000 prevents excessive computation. For Decision Trees, the Gini impurity criterion is used to evaluate splits, ensuring the selection of the most informative features, while a maximum tree depth of 20 prevents overfitting by limiting tree complexity. Random Forest employs an ensemble of 100 trees to enhance robustness, using Gini impurity for split decisions and setting the minimum samples per split and leaf to 2 and 1, respectively, to maintain flexibility in tree growth. In K-Nearest Neighbour, K is set to 5, balancing noise sensitivity and generalization, while the Minkowski distance metric is used for adaptive distance calculations. The uniform weight function ensures equal contribution from all neighbours in classification. These parameter choices collectively enhance the classifiers' generalization ability and efficiency, ensuring reliable performance on the CSE-CIC-IDS2018 dataset.

Table 4.2: ML classifier's parameter setting and their values used in the experiments

S. No.	ML Classifier	Parameter	Parameter Value
1.	SVM	Penalty	12
		Kernel	Linear
		Tolerance for stopping criteria	0.001
		Max iterations	1000
2.	DT	Splitting criterion	Gini
		Split strategy at each node	Best
		Min samples split	2
		Max tree depth	20
3.	RF	Number of trees	default = 100
		Splitting criterion	Gini
		Min samples split	2
		Min sample leaf	1
4.	K-NN	Number of neighbours (K)	5
		Weight function	Uniform
		Distance metric	Minkowski

Initially, after cleaning the dataset, we scaled it using the Min-Max scaler function that transformed the value of the data points to a standard scale ranging within [0, 1]. Consequently, two dimensionality reduction techniques, PCA and LDA, were performed on the datasets. Table 4.3 presents the number of original features in the dataset along with the number of transformed features or dimensions using these dimensionality reduction techniques. The criterion to minimise the number of features in our experiment is based on the cumulative variance and number of classes in the dataset for PCA and LDA, respectively. In preliminary experiments, detection accuracy was tested at various cumulative variance thresholds for PCA, including 80%, 85%, 90%, and 95%. The results showed that accuracy was 7-15% lower at cumulative variances between 80% and 90% compared to 95%. Consequently, a cumulative variance of 95% was chosen as the selection criterion for PCA. With this threshold, the original set of 80 features was reduced to 10 features or dimensions. However, LDA reduced the feature

set to 14 based on the selection criterion of $C-1$, where $C=15$ represents the number of attack classes in the dataset. It was observed that the use of SMOTE had a slight impact on dimensionality reduction, particularly with LDA. As a result, the number of dimensions remained 10 for PCA and 14 for LDA in both the original and balanced datasets, as presented in Table 4.3.

Table 4.3: Number of features before and after performing Dimensionality Reduction

Dataset		**Number of Features/ Dimensions**
Original		80
Dimensionality-reduced dataset	PCA	10
	LDA	14

Further, the binary classification of network traffic was analyzed using the accuracy, precision and F1 score evaluation metrics obtained by performing three cross-validation strategies, namely 5-fold CV, 10-fold CV, and LOOCV. Table 4.4 highlights the model's performance for binary classification of network data into benign and attack classes, along with the DR methods based on the accuracy (%) and F1-score (%) in the original dataset and for the balanced dataset in Table 4.5. Considering the dimensionality reduction methods, the results indicate that classifiers exhibit improved intrusion detection efficacy when the dataset's dimensionality is reduced using PCA and/or LDA. Using the LOOCV strategy, the highest accuracy for detecting network intrusions among all combinations of classifiers and dimensionality reduction methods was achieved by the PCA–RF model, reaching 99.91% for the original dataset and 99.94% for the balanced dataset. The bold values indicate the maximum performance achieved by the respective classifier.

Table 4.4 presents a performance comparison of ML classifiers using different dimensionality reduction techniques across three cross-validation strategies. It can be observed that, overall, Random Forest with PCA and LDA achieved the highest accuracy and F1-score, making it the most effective classifier. It is also evident that dimensionality reduction improves performance for all classifiers, with PCA performing the best. Also, among cross-validation strategies, LOOCV consistently yielded the best results, followed by the 10-Fold CV. RF with PCA using LOOCV achieved the highest accuracy of 99.91% and F1-score of 99.70%, making it the best-performing configuration. These

Table 4.4: Performance analysis of ML classifiers using three cross-validation strategies on the CSE-CIC-IDS2018 dataset

Classifier	DR method	Accuracy (%)			F1-Score (%)		
		5-Fold CV	10-Fold CV	LOOCV	5-Fold CV	10-Fold CV	LOOCV
SVM	WDR	90.65	93.18	93.39	90.47	92.22	92.99
	PCA	95.53	97.76	98.88	94.24	97.70	98.49
	LDA	94.99	97.77	98.48	94.20	96.93	98.17
DT	WDR	89.89	91.85	92.44	88.95	90.90	92.23
	PCA	94.44	96.23	98.88	94.76	95.99	98.39
	LDA	93.82	96.11	98.35	93.36	97.88	98.01
RF	WDR	94.79	96.04	96.99	94.59	95.16	95.41
	PCA	98.65	99.20	**99.91**	98.39	99.09	**99.70**
	LDA	98.15	98.89	99.62	98.01	98.27	99.59
K-NN	WDR	92.40	94.98	95.85	92.13	94.61	95.38
	PCA	97.90	99.01	99.67	97.69	98.49	99.12
	LDA	97.55	98.21	98.97	96.28	98.10	98.73

observations can be perceived from Fig. 4.5. These results highlight the importance of dimensionality reduction in improving classification performance, particularly for SVM and DT, while RF with PCA remains the optimal choice for anomaly detection in the CSE-CIC-IDS2018 dataset.

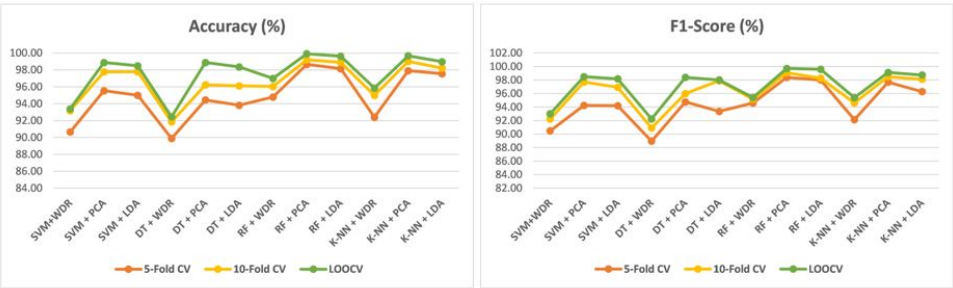

Fig. 4.5: Performance comparison for the combination of ML classifiers, dimensionality reduction techniques, and cross-validation strategies based on accuracy (%) and F1-score (%).

Table 4.5 presents the performance of ML classifiers on the SMOTE-balanced CSE-CIC-IDS2018 dataset using three cross-validation strategies. Across all classifiers, dimensionality reduction using PCA and LDA significantly improves performance com-

pared to WDR. The RF classifier demonstrates the highest accuracy and F1-score across all cross-validation methods, with PCA and LDA yielding exceptional results, achieving an accuracy of 99.94% and an F1-score of 99.78% with LOOCV. Overall, it was observed that PCA and LDA enhance classification accuracy and robustness, with RF emerging as the best-performing model on the SMOTE-balanced dataset.

Table 4.5: Performance analysis of ML classifiers using three cross-validation strategies on the SMOTE- balanced CSE-CIC-IDS2018 dataset

Classifier	**DR method**	**Accuracy (%)**			**F1-Score (%)**		
		5-Fold CV	**10-Fold CV**	**LOOCV**	**5-Fold CV**	**10-Fold CV**	**LOOCV**
SVM	WDR	90.72	93.22	93.41	90.54	92.26	93.01
	PCA	95.55	97.78	98.88	94.26	97.72	98.49
	LDA	95.06	97.77	98.56	94.27	96.98	98.27
DT	WDR	89.90	91.87	92.44	88.95	90.90	92.29
	PCA	94.45	96.26	98.88	94.77	96.00	98.44
	LDA	93.84	96.12	98.36	93.38	97.90	98.08
RF	WDR	94.83	96.06	97.01	94.63	95.18	95.43
	PCA	98.75	99.29	**99.94**	98.49	99.18	**99.78**
	LDA	98.21	98.93	99.64	98.07	98.31	99.61
K-NN	WDR	92.40	95.01	95.85	92.17	94.67	95.42
	PCA	97.94	99.03	99.69	97.73	98.51	99.14
	LDA	97.62	98.26	98.99	96.35	98.15	98.75

We have also investigated the relative performance of 24 combinations of 4 ML classifiers and 3 dimensionality reduction techniques across 3 cross-validation strategies, using a robust ranking mechanism [149] on both datasets. The ranking is performed using the average gain in the performance in comparison to the least performance accuracy (A_L) attained by the rest of the combinations of methods. For n number of combinations of classifiers and dimensionality reduction and without dimensionality reduction methods, the average percentage gain in accuracy (G_i) for the combination *i* is evaluated as:

$$G_i = \frac{1}{n} \sum_{c=1}^{n} \frac{acc_{cd} - A_L}{A_L} X100 \tag{4.57}$$

where acc_{cd} is the intrusion detection accuracy achieved for the model with i^{th} combination and classifier c.

Fig. 4.6 demonstrates the ranking of all combinations by sorting the relative accuracy of gain values in descending order. Subfigure (a) represents results on the original dataset, while subfigure (b) depicts results on the SMOTE-balanced dataset. The findings indicate that the relative accuracy gain for the combination of the Random Forest and PCA technique is higher than any other combination under consideration. The SMOTE-balanced dataset generally improves accuracy, demonstrating the effectiveness of handling class imbalance in intrusion detection. It is also evident that the performance gain (%) for any classifier without dimensionality reduction gives the worst performance than the one where dimensionality reduction is used, either PCA or LDA. It is also evident that the classifiers perform better with PCA rather than LDA.

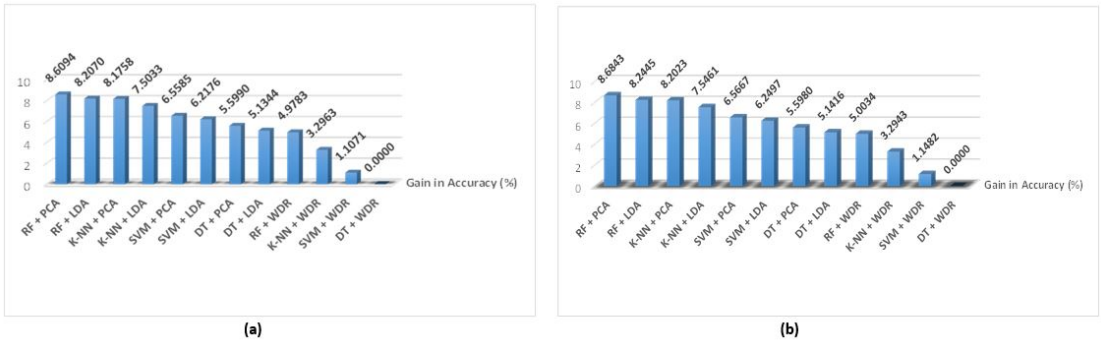

Fig. 4.6: Ranking of the combination of ML classifiers and dimensionality reduction techniques based on Performance Gain in Accuracy (%) on (a) Original CSE-CIC-IDS2018 and (b) SMOTE-balanced CSE-CIC-IDS2018

Table 4.6 summarizes the comparative analysis of the proposed framework's binary classification performance with the existing models on the CSE-CIC-IDS2018 dataset. The proposed PCA-RF model outperforms all other methods, achieving the highest accuracy of 99.94% and F1-score of 99.78%, surpassing even advanced deep learning-based techniques like XIDINTFL-VAE and EAFS-RF. These results highlight the effectiveness of PCA for feature reduction combined with RF for classification, leading to superior performance in anomaly-based network intrusion detection.

Table 4.6: Performance analysis of the PCA-RF binary classification with existing models on CSE-CIC-IDS2018 dataset

Reference	Model	Accuracy (%)	F1-Score (%)
[49]	GB	99.32	-
[44]	Ensemble Learning model	98.80	-
[54]	DSSTE+ miniVGGNet	96.99	97.04
[47]	one-class SVM	88.98	-
[53]	SVM	75.59	-
[150]	CNN	73.83	-
[151]	EAFS-RF	99.04	98.33
[152]	RF	99.60	-
[55]	XIDINTFL-VAE	99.89	97.05
Proposed framework	**PCA – RF**	**99.94**	**99.78**

4.4.3 Results of Clustering-Based Network Attack Analysis

In this subsection, after the binary classification of the dataset into benign and attack instances, the latter is further clustered using the proposed mP$_{ic}$FC approach to identify different types of network attacks present in the dataset. To assess the effectiveness of the proposed clustering technique, we compare its performance with state-of-the-art clustering approaches such as K-Means, FCM, IFCM, and PFC, which are widely used in network intrusion detection. The evaluation is conducted on both the original and class-balanced datasets obtained after the second phase of the framework. The hyperparameter tuning details for the clustering phase are presented in Table 4.7, with cluster centroids initialized randomly. The comparison aims to determine the efficiency of mP$_{ic}$FC in accurately grouping attack types, ensuring better anomaly detection in cybersecurity applications.

As presented in Table 4.7, each model is configured with six clusters (C = 6) to categorize different attack types. K-Means uses a tolerance (Tol) of 0.0001, while FCM, IFCM, PFC, and mP$_{ic}$FC share common parameters such as the fuzziness parameter (m = 2), convergence criterion (ϵ = 0.0001), and a maximum iteration limit of 1000 steps. Additionally, IFCM, PFC, and the proposed mP$_{ic}$FC incorporate a parameter δ, ranging from 0 to 2 with a step size of 0.15, allowing enhanced flexibility in clustering. The

Table 4.7: Parameter setting for the state-of-the-art and proposed clustering approach used in the experiments

S. No.	Clustering Model	Parameters
1.	K-Means	Number of clusters, C = 6, Tol = 0.0001
2.	FCM	C = 6, m = 2, ϵ = 0.0001, maxSteps = 1000
3.	IFCM	C = 6, m = 2, ϵ = 0.0001, δ = [0-2] with a step size of 0.15, maxSteps = 1000
4.	PFC	C = 6, m = 2, ϵ = 0.0001, δ = [0-2] with a step size of 0.15, maxSteps = 1000
5.	Proposed mP$_{ic}$FC	C = 6, m = 2, ϵ = 0.0001, δ = [0-2] with a step size of 0.15, maxSteps = 1000

inclusion of these models facilitates a comparative performance analysis, with the proposed mP$_{ic}$FC aiming to improve clustering efficiency in network intrusion detection.

We tuned all clustering models using internal validity indices and then reported any external, label-based metrics only after selection to prevent bias. For the fuzzy models, FCM, IFCM, PFC, and mP$_{ic}$FC, we minimized the XB and monitored the PC and PE values. For K-means, the hyperparameters were chosen by minimizing inertia, that is, the sum of squares within the cluster. We further verified consistency by computing PC and PE from the 0 or 1 cluster memberships. Tol = $1e^{-4}$ was used because it avoided a few early stops observed with $1e^{-3}$. For the fuzzy models, the fuzziness parameter *m* is set to $m = 2$, following the PFC baseline [115] that mP$_{ic}$FC extends. The $\epsilon = 10^{-4}$ avoided the occasional premature halts seen with 10^{-4} and δ is tested in a range of [0-2] as very small δ produced over-confident memberships, that is, higher PE while very large δ over-penalised refusal, higher XB.

Initially, to validate the quality of the clustering techniques, a performance comparison is done using three clustering validity indices, PC, PE, and XB. The optimal values for these metrics are stated in Tables 4.8 and 4.9. It can be observed that the proposed mP$_{ic}$FC clustering approach outperformed the state-of-the-art methods as it achieved the maximum value of PC at 0.8796 and minimum values of PE and XB at 0.1288 and 0.2892 respectively for original data samples. The proposed fuzzy clustering approach for the balanced dataset achieved a peak PC value of 0.8801, along with minimum PE and XB values of 0.1217 and 0.2879, respectively. This demonstrates the superior clustering capability of the mP$_{ic}$FC method compared to other models.

The detailed performance evaluation of the mP$_{ic}$FC in detecting the various network attack categories, in terms of accuracy, precision, FPR, FNR and their respective average

Table 4.8: Performance comparison using clustering validity indices for CSE-CIC-IDS2018

Model	PC	PE	XB
K-Means [28]	0.7908	0.4637	0.5241
FCM [122]	0.8315	0.3212	0.4388
IFCM [124]	0.8263	0.4146	0.4941
PFC [115]	0.8689	0.1875	0.2946
mP$_{ic}$FC	**0.8796**	**0.1288**	**0.2892**

Table 4.9: Performance comparison using clustering validity indices for SMOTE-balanced CSE-CIC-IDS2018

Model	PC	PE	XB
K-Means [28]	0.8027	0.4499	0.4911
FCM [122]	0.8454	0.3076	0.4003
IFCM [124]	0.8318	0.4022	0.4776
PFC [115]	0.8699	0.1855	0.2899
mP$_{ic}$FC	**0.8801**	**0.1217**	**0.2879**

values computed, are summarized in Tables 4.10 and 4.11, with mP$_{ic}$FC securing an average detection accuracy of 89.74%, average precision of 88.80%, an average FPR of 0.000333, and average FNR of 0.009833 for the original dataset and average detection accuracy of 89.98%, average precision of 89.07%, an average FPR of 0.000210, and average FNR of 0.008816 for the SMOTE-balanced dataset. The results presented are inclusive of the various sub-attack classes detected by the clustering methods under consideration.

It can also be observed that on the original data, mP$_{ic}$FC produces an average FPR = 0.000333, which is about 3.3 false alarms per 10,000 benign flows (approximately 1 in 3,003). The false positives are concentrated on two attacks, DDoS and Botnet, each at an FPR of 0.001, indicating around 1 in 1000 benign flows are labelled as that class. After performing SMOTE, the average false-alarm rate drops to 0.000210, that is, 2.1 per 10,000 benign flows or approximately 1 in 4,762, which is a 37% reduction. The worst case FPR also improves from 0.001 to 0.0006, for both DDoS and Botnet attacks, making it a 40% decrease. Overall, with FPR already low, class balancing

Table 4.10: Performance evaluation of mP$_{ic}$FC for detecting network attacks

Network Attack Class	Acc.	Prec.	FPR	FNR
DDoS	0.889	0.875	0.001	0.0120
DoS	0.899	0.893	0.000	0.0090
Bruteforce	0.898	0.891	0.000	0.0110
Botnet	0.892	0.886	0.001	0.0100
Infiltration	0.899	0.891	0.000	0.0090
Web attack	0.899	0.892	0.000	0.0080
Average	**0.8974**	**0.8880**	**0.000333**	**0.009833**

Table 4.11: Performance evaluation of mP$_{ic}$FC for detecting network attacks for SMOTE-balanced dataset

Network Attack Class	Acc.	Prec.	FPR	FNR
DDoS	0.892	0.877	0.0006	0.0100
DoS	0.902	0.895	0.0000	0.0084
Bruteforce	0.901	0.893	0.0000	0.0077
Botnet	0.895	0.887	0.0006	0.0100
Infiltration	0.902	0.896	0.0000	0.0088
Web attack	0.902	0.894	0.0000	0.0080
Average	**0.8998**	**0.8907**	**0.000210**	**0.008816**

suppresses residual DDoS or Botnet peaks, reducing both the average and the worst-case FPR without degrading other classes.

Across original data, mP$_{ic}$FC maintains stable performance across all attack classes. FNR values are consistently below 1.2%, ensuring that only a very small fraction of attacks go undetected. In particular, Web attacks exhibit the lowest FNR of 0.8%, while DDoS shows a slightly higher FNR of 1.2%. However, balancing with SMOTE further improves consistency across classes. All FNR values drop to below 1%, demonstrating that balancing enhances sensitivity, especially for minority attacks such as Infiltration and Web attacks. The near-zero FPR combined with the very low FNR confirms mP$_{ic}$FC's ability to minimize both false alarms and missed detections simultaneously.

Comparative performance analysis of all the clustering methods for the detection of different network attack types based on accuracy, precision, FPR, and FNR is illustrated in Figs. 4.7 and 4.8. It can be observed that the hard clustering method, K-Means, had the lowest performance in terms of the said metrics in the case of detecting all the attack types. However, it is evident that in detecting all the attacks, the proposed mP$_{ic}$FC method secured the maximum detection accuracy and precision as well as the lowest FPR and FNR, indicating superiority over other state-of-the-art methods.

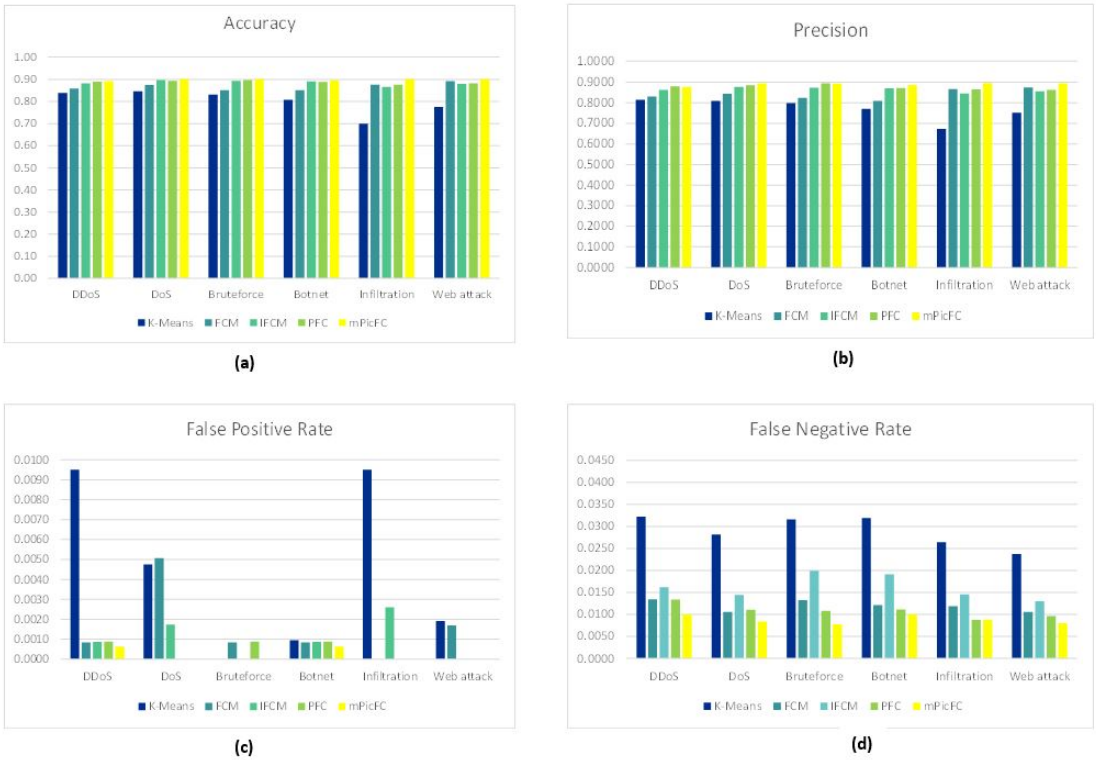

Fig. 4.7: Performance comparison of clustering techniques for detecting all network attacks based on (a) Accuracy, (b) Precision, (c) False Positive Rate, and (d) False Negative Rate

Considering the average values of the above four metrics for all the clustering methods, a comparison of their performance in detecting various network attacks is demonstrated in Tables 4.12 and 4.13 and Fig. 4.9. From the results obtained, it can be observed that K-Means had the worst performance among the rest of the fuzzy clustering methods, achieving the lowest average accuracy of 89.74% and the highest average FPR and FNR of 0.0046 and 0.0320, respectively, across both sets of data, missing a substantial portion of attacks and a frequent number of false alarms. On the unbalanced original

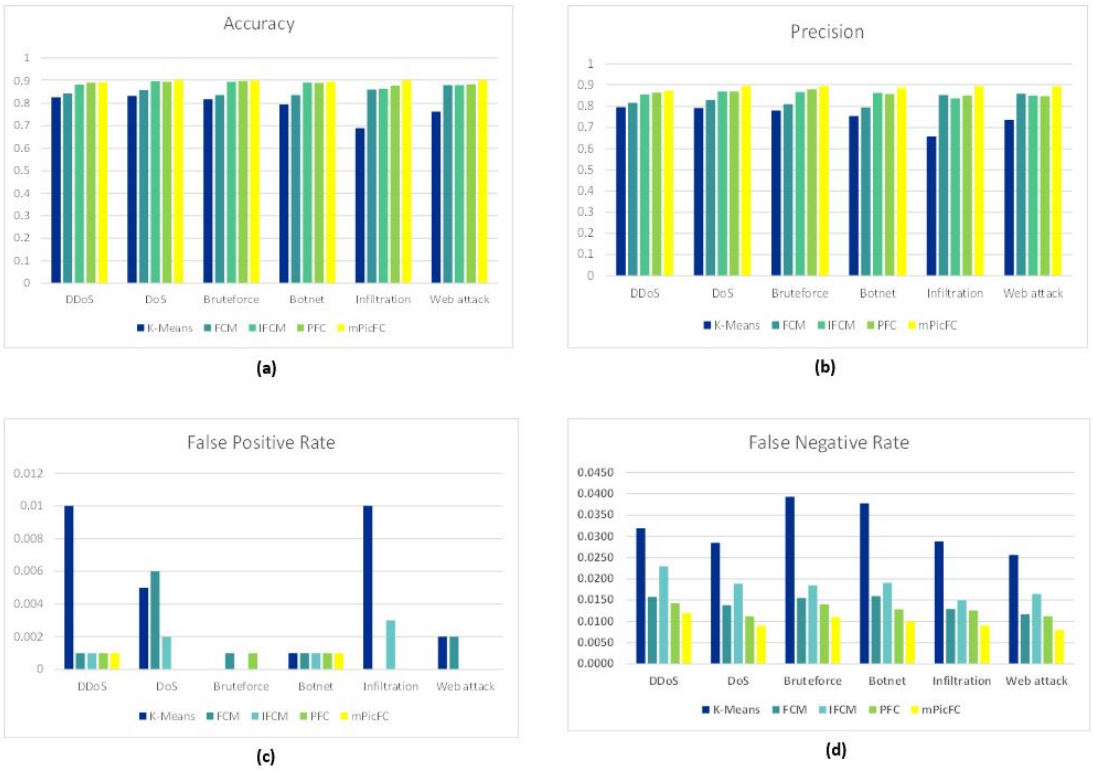

Fig. 4.8: Performance comparison of clustering techniques for detecting all network attacks on balanced dataset based on (a) Accuracy, (b) Precision, (c) False Positive Rate, and (d) False Negative Rate

dataset, FCM and IFCM reduce the FNR to around 1.4–1.8% and the FPR below 0.2%, while PFC further improves it to 1.2%. However, dataset balancing improves detection across all models by reducing FNR while preserving low FPR. While K-Means continues to lag, the fuzzy clustering methods, FCM, IFCM, and PFC, consistently reduce false negatives, with FNR dropping to a range of 1.0–1.6%, and FPR also remaining very low at less than 0.15%.

It can also be observed from Fig. 4.9 that the false positive rate and the false negative rate drastically declines as we move from hard to soft or fuzzy clustering methods, indicating that fuzzy clustering is more efficient in detecting network attacks than hard clustering, and the proposed clustering method is efficient than the rest.

Subsequently, among all the clustering methods, across both datasets, the proposed mP$_{ic}$FC clustering model achieved the highest average detection accuracy and precision of 89.98% and 89.07%, respectively, and the lowest FPR and FNR of 0.000210

Table 4.12: Performance comparison of clustering techniques for the CSE-CIC-IDS2018 dataset based on average values of accuracy, precision, FPR, and FNR

Model	Acc.	Prec.	FPR	FNR
K-Means [28]	0.7873	0.7535	0.0046	0.0320
FCM [122]	0.8522	0.8283	0.0018	0.0142
IFCM [124]	0.8822	0.8587	0.0011	0.0185
PFC [115]	0.8858	0.8630	0.0005	0.0127
mP$_{ic}$FC	**0.8974**	**0.8880**	**0.000333**	**0.009833**

Table 4.13: Performance comparison of clustering techniques for the SMOTE-balanced CSE-CIC-IDS2018 dataset based on average values of accuracy, precision, FPR, and FNR

Model	Acc.	Prec.	FPR	FNR
K-Means [28]	0.8005	0.7699	0.0044	0.0290
FCM [122]	0.8679	0.8417	0.0015	0.0120
IFCM [124]	0.8856	0.8644	0.0010	0.0162
PFC [115]	0.8883	0.8772	0.0004	0.0108
mP$_{ic}$FC	**0.8998**	**0.8907**	**0.000210**	**0.008816**

and 0.008816, respectively, outperforming the rest of the state-of-the-art techniques in detecting the types of network attacks. The results also suggest that on the original dataset, by achieving the lowest FPR and FNR, the proposed method minimizes both false alarms and missed attacks simultaneously, which is a particularly challenging trade-off in intrusion detection, while maintaining negligible false alarms, strengthening its reliability across minority attack classes. It is vital to note that these comparative analyses are performed with existing methods on the intrusion detection dataset CSE-CIC-IDS2018 only.

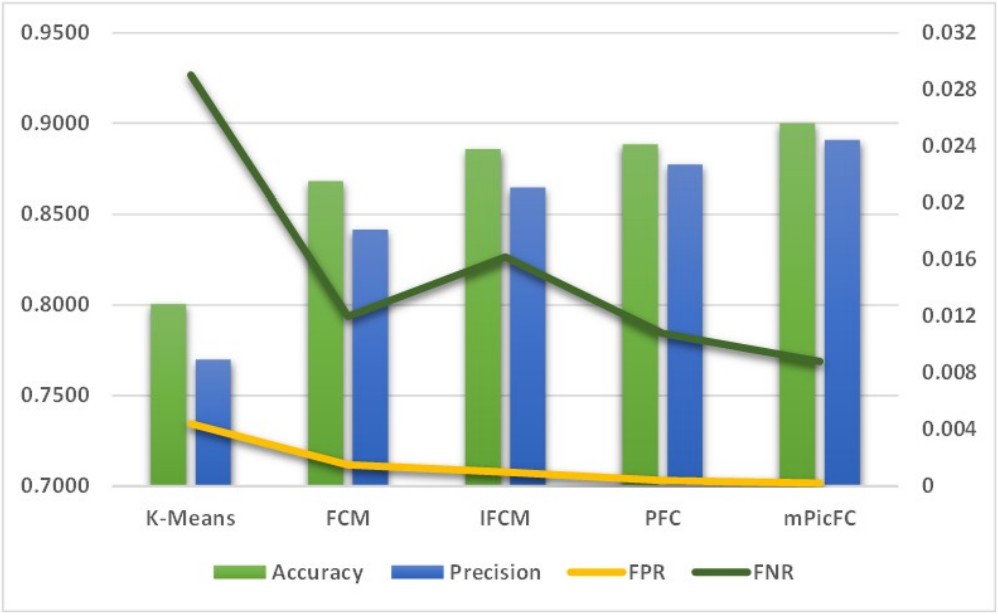

Fig. 4.9: Performance comparison of proposed mP$_{ic}$FC with other clustering methods based on average values of accuracy, precision, FPR, and FNR

4.4.4 Statistical Significance

This subsection discusses the statistical validation of the proposed mP$_{ic}$FC by performing the Friedman test and post-hoc Wilcoxon signed-rank test with Holm correction. The test is evaluated by comparing the five clustering methods, K-Means, FCM, IFCM, PFC, and the proposed mP$_{ic}$FC, using the validity indices, PC, PE, and XB on both original and SMOTE-balanced datasets. We also test them based on external metrics, including accuracy, precision, and FPR across attack classes, on both datasets.

To run a single non-parametric test across different metrics, we put everything on a loss scale, that is, PC, accuracy and precision, which is higher values, were flipped to $1 - value$. PE, XB, and FPR are unchanged. Each (metric, dataset) pair constitutes a repeated-measures block for the non-parametric tests.

For the Friedman test, the null hypothesis (H_0) states that all methods perform equivalently, and the alternative hypothesis (H_1) indicates that at least one method performs significantly differently. For $k = 5$ clustering algorithms and $N = 12$ number of blocks, a cumulation of six for internal validity and six for external performance over both datasets, r_{ij} the rank of the method j in i^{th} block, the Friedman statistic is computed

as presented in Eqn. 3.11. For this combined set, the effect is more potent than when considering internal and external performances, as the evaluated F-statistic is 21.60 for a *p-value* of $2.98e^{-9}$, keeping $\alpha = 0.05$. The average ranks of the Friedman test for all clustering models across all evaluation blocks are presented in Table 4.14, where a lower rank indicates better overall performance.

Table 4.14: Average Ranking of the Clustering Methods using the Friedman Test

Clustering Method	Average Ranking
mP$_{ic}$FC	1.09
PFC	2.05
IFCM	3.50
FCM	3.60
K-Means	5.00

Table 4.15: Post-hoc Wilcoxon-Holm Test Comparing mP$_{ic}$FC with Baseline Methods

Comparison	p-value	Reject
mP$_{ic}$FC vs K-Means	0.000443	Y
mP$_{ic}$FC vs FCM	0.000327	Y
mP$_{ic}$FC vs IFCM	0.000211	Y
mP$_{ic}$FC vs PFC	0.000109	Y

Y = Yes, N = No.

The clear separation of between the mP$_{ic}$FC model and other baseline methods suggests a pronounced and stable advantage for the proposed approach, consistent with the expectation that it handles overlapping or uncertain boundaries better than crisp partitioning. The pair-wise post-hoc Wilcoxon-Holm correction test revealed that the proposed method is significantly superior to all four baseline methods, presented in Table 4.15 along with Holm-adjusted p-values.

4.5 Chapter Summary

This chapter presents an anomaly-based Network Intrusion Detection System framework that addresses the high dimensionality challenge and class imbalance issue in in-

trusion detection datasets using machine learning and the modified Picture Fuzzy Clustering (mP$_{ic}$FC) approach. Evaluated on the CSE-CIC-IDS2018 dataset, the framework first applies Synthetic Minority Oversampling Technique (SMOTE) to handle the class imbalance problem, ensuring improved detection of rare attacks such as Web Attacks and Infiltration attempts. Following this, binary classification is performed to categorize network traffic into benign and attack instances using four ML classifiers (SVM, Decision Tree, Random Forest, and K-NN) applied to a dimensionality-reduced dataset using PCA and LDA, where Random Forest with PCA achieves the highest accuracy of 99.94% and 99.78% F1-score, outperforming existing models. In the clustering phase, the proposed mP$_{ic}$FC approach further groups attack instances, demonstrating superior clustering accuracy compared to K-Means, FCM, IFCM, and PFC, as reflected by the highest Partition Coefficient (PC = 0.8801) and lowest Partition Entropy (PE = 0.1217) and Xie-Beni Index (XB = 0.2879) in the SMOTE-balanced dataset. The mP$_{ic}$FC model offers several advantages, including enhanced handling of uncertainty by incorporating an additional membership degree, that is, neutrality and hesitation, superior clustering quality, and robustness against noise and outliers through the Hypertangent Kernel function. Additionally, the integration of PCA and LDA reduces computational complexity, while the SMOTE-balanced dataset further enhances classification performance. With an average detection accuracy of 89.98% and the lowest false positive rate, FPR of 0.000210, the proposed model significantly enhances intrusion detection capabilities, making it a scalable and efficient solution for real-world cybersecurity applications. To ensure a controlled comparison, we constrained the evaluation to the CSE-CIC-IDS2018 dataset. This tight scope gives strong internal validity, and testing across additional datasets is deferred to future work. We also focus on the attack classes captured in this dataset under stable conditions, establishing a robust baseline for subsequent cross-dataset studies.

Chapter 5

Hybrid Incremental Learning-Based Real-time NIDS

This chapter presents a network intrusion detection framework for anomaly detection in network data streams that exhibit evolving characteristics. Addressing the challenge of dynamically evolving networks, this study proposes a novel hybrid incremental learning framework for anomaly-based intrusion detection, HIL-IDS.

5.1 Introduction

In various domains like cybersecurity, fraud prevention, social media, and health monitoring, detecting anomalies in data streams has become crucial. With the rapid increase in cyber threats, ensuring the integrity, confidentiality, and availability of networked systems is more important than ever. Traditional intrusion detection systems, which rely on static datasets and predefined attack signatures, struggle to detect evolving and sophisticated cyberattacks. This limitation arises because real-time network traffic exhibits non-stationary behaviour, where attack patterns continuously change over time. In real-time scenarios, data distribution varies over a period of time as the real-time data streams often have a non-stationary behaviour. Due to this, real-time IDS has become more essential as it helps detect and alert any security breaches, mitigate evolving threats, process information instantly, and identify any malicious activity without compromising network performance.

In order to limit such problems, this study proposes a HIL-IDS model, a novel intrusion detection system based on a hybrid incremental learning approach performed on real-time network data traffic. Initially, network traffic packets were ingested in real-time using a packet sniffer called Scapy [153], which further performed feature extraction from the raw network packets. The processed data were then fed to the Apache Kafka pipeline [154], an open-source platform for logging, processing, and monitoring the network traffic. The hybrid incremental learning model, which integrates both supervised and unsupervised machine learning approaches for anomaly detection, provides a robust and adaptive intrusion detection mechanism. Before deploying the incremental learning model on the raw packets, it was trained on the publicly available intrusion detection dataset, CSE-CIC-IDS2018, as a benchmark, which consists of modern network attack patterns, helping our proposed intrusion detection model fine-tune on a wide coverage of network attacks and preparing it to handle realistic scenarios.

The pre-trained Hoeffding tree classifier [155] performs incremental supervised learning on the incoming stream of network traffic. It is an incremental decision tree algorithm that continuously updates its structure based on new network traffic patterns. This ensures that the model remains adaptive, reducing the risk of outdated classification rules. Also, it predicts whether the incoming data packet is normal or malicious. For the unsupervised approach of the model, Isolation forest [156] and KMeans were utilized for anomaly detection, providing an additional layer of security by identifying novel attack patterns. Instead of simply flagging the network packets as normal or malicious, the model evaluates a confidence score from the ensemble of the two unsupervised methods to make a more accurate decision about the network flow. The incremental learning model combines the predictions of the supervised and unsupervised models, refines its output, and compares it against the predefined threshold to evaluate the network data packets. Each sample was processed incrementally, the models were updated, and the output was refined in real time. Using the Hoeffding tree, we model non-linear relationships, and the ensemble unsupervised method is more flexible in detecting non-linear anomalies, making our model more efficient. Also, the proposed model handles drift effectively as the supervised model is inherently adaptive, and the unsupervised ensemble is more sensitive to dynamic changes. By continuously learning from live network traffic, adapting to evolving attack vectors, and integrating both supervised and unsupervised learning methods, the proposed HIL-IDS framework establishes a highly effective,

adaptive, and intelligent approach to real-time intrusion detection.

The key contributions of the proposed model are summarized as:

- The proposed framework performs real-time analysis and processing of sniffed network packets for efficient data streaming using Apache Kafka.
- A novel Hybrid Incremental Learning intrusion detection system, HIL-IDS, using a supervised Hoeffding Tree classifier and an unsupervised ensemble of Isolation Forest and KMeans, is proposed for adaptive and real-time anomaly detection in the network.
- The proposed work evaluates a combined confidence score from supervised and unsupervised models for output refinement and improved detection accuracy.
- Adaptive drift detection is done by actively monitoring the network and further retraining the model to maintain its performance in the evolving network environment.

This chapter presents the proposed HIL-IDS model in section 5.2, followed by the experimental setup outlined in section 5.3 along with a comparative evaluation of the results and statistical analysis of the proposed approach with existing models. Finally, section 5.4 provides a summary of the chapter.

5.2 Proposed Framework

To effectively tackle the challenges of real-time anomaly detection in dynamic network environments, this study presents the Hybrid Incremental Learning Intrusion Detection System (HIL-IDS). The proposed framework seamlessly integrates supervised and unsupervised learning techniques to enable adaptive, real-time detection of cyber threats while addressing concept drift in network traffic. By leveraging incremental learning, the model continuously updates itself with evolving attack patterns, reducing the risk of outdated detection rules. The combination of supervised classification and unsupervised anomaly detection enhances accuracy by identifying both known and unknown threats. This section provides a detailed overview of the HIL-IDS architecture, including data

processing, real-time network traffic handling, and the hybrid incremental learning approach that enables efficient and intelligent intrusion detection.

Fig. 5.1 illustrates the comprehensive architecture of the proposed framework, developed in four phases: real-time network traffic ingestion and pre-processing, pipeline integration with Apache Kafka for real-time message streaming, anomaly detection by the hybrid incremental learning (HIL) model, and output refinement and prediction.

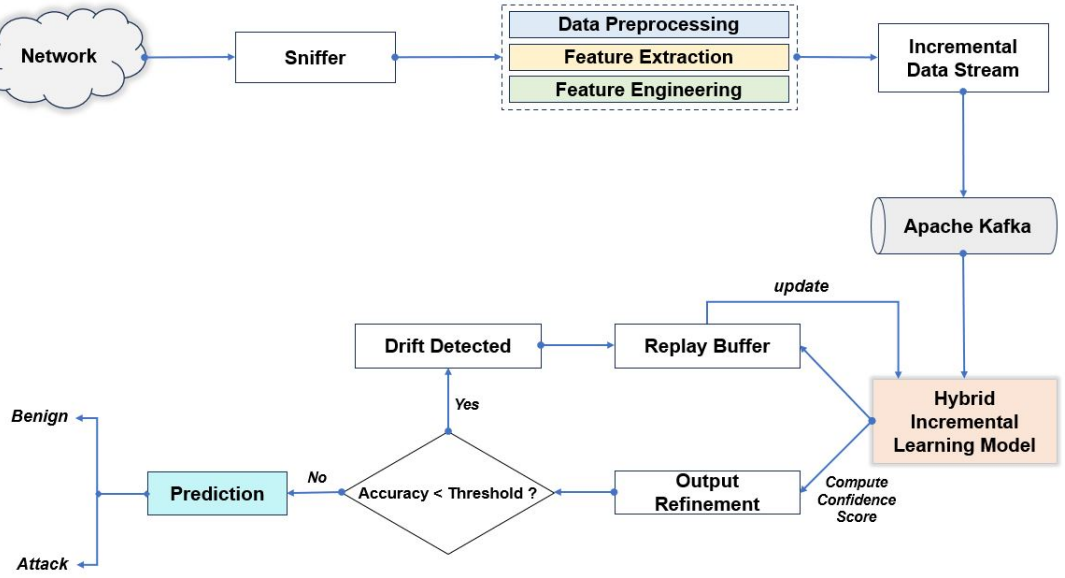

Fig. 5.1: Overall framework of the proposed HIL-IDS model

5.2.1 Network Traffic Ingestion

In the first phase of the framework, Scapy, a powerful packet manipulation tool, was utilized to sniff, ingest, and analyze real-time network packets. Following data ingestion, 17 key features—such as IP address, payload size, protocol, and port address—were extracted from the raw network packets. These extracted features were then pre-processed and cleaned while preserving data integrity to enhance prediction accuracy. Missing or null values were appropriately handled, and infinite values were transformed to a defined threshold to prevent skewing the model.

After data cleaning, feature engineering techniques were applied, including normal-

ization and dimensionality reduction. Min-Max scaling was employed to standardize feature values, improving model efficiency and mitigating the challenges of high dimensionality. This normalization technique was chosen as it preserves the relationships between data points without introducing latent bias [157]. Moreover, it is computationally efficient and less complex than alternative methods such as Z-score and sigmoidal normalization, as discussed in section 4.3.1.

As observed and concluded in the previous chapter, Principal Component Analysis (PCA) has proven to be an efficient dimensionality reduction technique; hence, to further optimize model performance, PCA was applied to the pre-processed and normalized dataset. PCA transformed the data into a lower-dimensional space while retaining its essential characteristics, thereby reducing computational overhead and enhancing the model's effectiveness in real-time anomaly detection.

5.2.2 Apache Kafka Pipeline Integration

Apache Kafka [158], an open-source distributed event-streaming platform, helps perform real-time data processing. It works on the concept of a Producer-Consumer messaging system and can handle high throughput data streams, making it highly efficient for detecting potential threats in the network [159, 160]. It can collect, store, and route network data packets from various sources and, via this pipeline, feeds them to the intrusion detection models via a robust, scalable pipeline [161]. Kafka ensures fault tolerance and high availability by replicating data across multiple brokers. This guarantees uninterrupted network monitoring even in the event of node failures. Kafka efficiently manages growing network traffic by distributing incoming data across multiple partitions and brokers, ensuring high performance without degradation. Fig. 5.2 demonstrates the working of the Apache Kafka Pipeline in our proposed approach.

In the proposed framework, a Kafka producer publishes the transformed data to a designated Kafka topic, a logical channel for message transfer, in an incremental streaming fashion in real-time. The Kafka topics are split into partitions, enabling parallel processing and load balancing. The data is then routed to the Kafka message broker cluster, a server that stores and manages the messages.A Zookeeper ensemble oversees the Kafka cluster, handling broker coordination, topic configurations, leader election, and failure detection, thus ensuring high availability and resilience of the message pipeline.

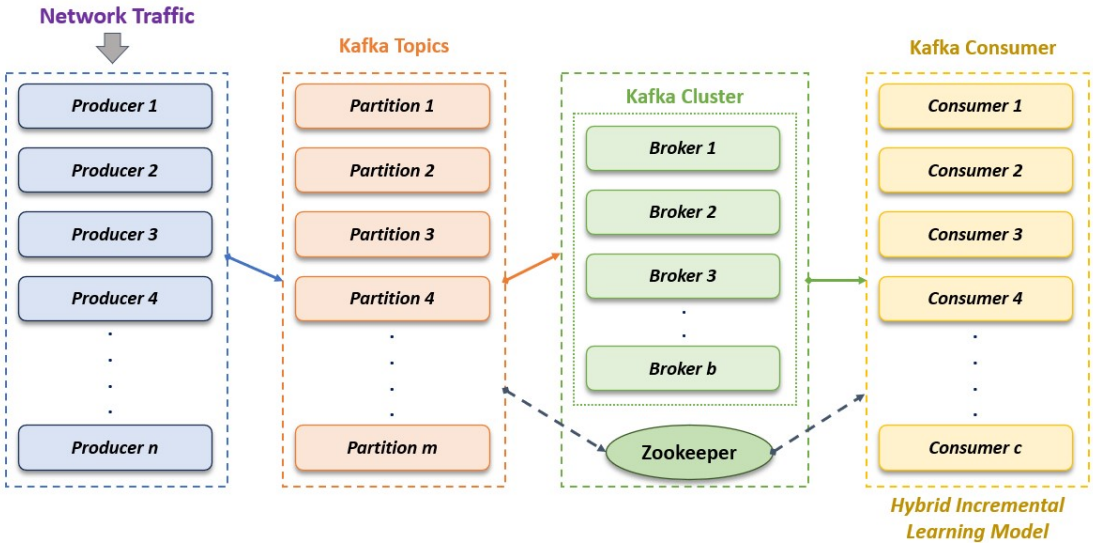

Fig. 5.2: Overall framework of the proposed HIL-IDS model

On the consumer side, a Kafka consumer continuously reads the assigned topic partitions, fetching the most recent batches of transformed feature vectors representing live network traffic. This enables the detection system to adapt to evolving traffic patterns, including anomalies and intrusions, in real-time. In the proposed approach, the HIL model serves as the Kafka consumer. It ingests the streamed data, processes it incrementally to reflect concept drift, and performs dynamic intrusion detection without requiring retraining from the beginning.

Since a real-time IDS requires continuous data processing, and the producer-consumer model ingests data indefinitely, termination strategies are necessary to control the consumer's execution. These strategies include setting a consumer timeout, imposing predefined termination conditions, manually stopping the consumer, or allowing it to halt when the Kafka topic is depleted, that is, no messages remain to consume. In this study, if messages or data packets are available, the consumer continues ingesting data until it either reaches the timeout or the end of the data packet stream. If no data packets are available, the consumer stops after a timeout period if no new packets arrive. In this study, based on the previous related works, the Kafka pipeline is assumed to deliver traffic reliably, without packet loss or timestamp misalignment, so that incremental updates remain meaningful.

5.2.3 Proposed Hybrid Incremental Learning Model

The proposed intrusion detection framework based on the hybrid incremental learning model, HIL-IDS, integrates a supervised ML model, the Hoeffding Tree classifier, and an amalgamation of two unsupervised ML techniques, isolation forest and KMeans clustering. The Hoeffding tree, also known as the Very Fast Decision Tree (VFDT), is an incremental decision tree learning model that is efficient for streaming data [162]. Due to its capability to handle continuous and large amounts of streaming data incrementally, it can adapt to the changes in the data distributions, which is useful for concept drift handling. Also, hoeffding trees are fast and memory-efficient, making them suitable for high-speed network traffic analysis where low latency is essential and is, therefore, chosen for our study.

For unsupervised learning, isolation forest, an anomaly detection algorithm, isolates the data points using random partitioning. It is suitable for scenarios where anomalous patterns are present in the network by isolating them from the majority of the data. It assigns an anomaly score to each data sample based on its average path length in the forest. If the average path is shorter, the more unusual the data sample is, the higher the anomaly score will be. Due to its efficiency in handling imbalanced data and supporting fast real-time processing, it is incorporated into the proposed model. KMeans clustering model, on the other hand, is effective in identifying the "normal" patterns in the dataset and also identifies clusters of intrusions. It is designed to handle large datasets efficiently, making it suitable for real-time applications.

Together, they can differentiate between isolated outliers and anomalous clusters of data. KMeans helps refine the results from Isolation Forest by filtering anomalies based on cluster characteristics, which helps reduce the false positive rate. This combination ensures adaptability to dynamic data as KMeans adapts to changing traffic patterns by re-clustering data periodically while isolation forest updates anomaly scores incrementally, increasing the robustness of the model. Thus, fusion of incremental and ensemble learning, that is, the integration of Hoeffding Tree for incremental learning with an unsupervised anomaly detection ensemble is a novel and efficient approach, ensuring both adaptability and robustness.

Initially, this hybrid incremental learning model is trained on the pre-processed open-source network intrusion detection dataset, CSE-CIC-IDS2018, which consists

of novel modern network attacks that help train our model to identify newer attack patterns. Scores evaluated from the supervised and unsupervised models are combined and evaluated, and these weighted scores are further used for incremental output refinement and prediction.

5.2.4 Output Refinement

In the final phase, the outcome of the hybrid of supervised and unsupervised models was computed and further processed to predict whether the data packet is anomalous or not. Based on the previously learned patterns from the existing dataset, the pre-trained Hoeffding tree predicts and provides a label for whether the incoming packet is benign or an attack. Anomaly scores, normalized to a range of 0–1, were computed for both Isolation Forest and KMeans clustering. For the Isolation Forest, a higher anomaly score indicates more anomalous behavior in the data packet, while for KMeans, larger distances to the cluster centers signify a higher anomalous pattern. The combination of supervised and unsupervised techniques ensures a more resilient anomaly detection approach. While the Hoeffding Tree excels at recognizing previously known attack patterns, Isolation Forest and KMeans help uncover novel or evolving anomalies, making the model adaptive to real-world cyber threats. The anomaly scores from both methods are averaged to create a combined anomaly score.

The model then computes a weighted score by combining supervised predictions with unsupervised anomaly scores for the incoming data, evaluated as:

$$weighted\_score = (sup\_pred * sup\_weight) + (anomaly\_score * unsup\_weight) \tag{5.1}$$

where *sup_pred* is the confidence score evaluated from the supervised model, and *anomaly_score* is the confidence score from the unsupervised anomaly detection model. The *sup_weight* determines the relative importance of the supervised prediction, while *unsup_weight* is computed as (*1 – sup_weight*). The formula combines the scores of both models, and the weighted output is computed.

To determine the optimal threshold for classifying a data packet as anomalous or benign, multiple threshold values (0.60, 0.65, 0.70, 0.75, and 0.80) were tested, and their corresponding False Positive Rate (FPR) vs. False Negative Rate (FNR) was analyzed.

The results showed the most promising trade-off at a threshold of 0.65, effectively balancing detection accuracy and minimizing false alerts. If the *weighted_score* for a data packet exceeds this threshold, it is classified as anomalous; otherwise, it is benign.

The model also monitors the accuracy of the supervised model on real-time data increments and compares it with the initial accuracy. If performance drops below a predefined threshold, a concept drift is flagged, requiring the model to adapt to new data distributions using recent samples stored in the replay buffer. The replay buffer stores recent samples after predictions are made, ensuring that predictions are computed before a sample is added for potential model updates or retraining. By detecting drift or changes in data distribution, the system can trigger model updates to maintain accuracy and prevent performance degradation over time.

5.3 Experimental Results and Analysis

This section highlights the key outcomes of the proposed real-time intrusion detection system, which is based on a hybrid incremental learning model that ensembles supervised and unsupervised ML models. In the proposed HIL-IDS model, for the network data packets sniffed, initially, the features were extracted using Scapy, pre-processed and then standardized using Min-Max scaling. After scaling the data, dimensionality reduction was performed using the powerful PCA technique. 17 key features were extracted from the input data stream. The criterion to minimise the number of features in our experiment is based on the cumulative variance and number of classes in the dataset. A cumulative variance of 95% was chosen as the selection criterion for PCA. With this threshold, the original set of 17 features was reduced to 12 features or dimensions.

The performance of the HIL-IDS model was evaluated using standard classification metrics, including accuracy, F1 score, and FPR. The experiments were performed on a computer system having a Windows 10 system with an Intel i7 (9^{th} generation) processor and 32GB RAM, using 64-bit Spyder with the Python programming language.

The various parameter settings for the machine learning models used in the experiments are detailed in Table 5.1. The Hoeffding Tree model utilizes three parameters: *grace_period*, which determines the number of instances seen before a split attempt. A smaller value leads to faster learning but might increase the risk of overfitting. Thus, it is set to a larger value of 100. *split_confidence* set to 1e-5, which controls the confi-

dence level for splitting nodes, a higher value leads to more conservative splitting; and *tie_threshold* of 0.05, which serves as the threshold for considering splits as ties. The Isolation Forest method employs estimators set to 200, indicating the number of base estimators (or trees) in the ensemble, and contamination set to 0.1, representing the proportion of anomalies or outliers in the dataset. The KMeans clustering algorithm is configured with *Number of clusters* set to 7, and a *mini_batch_size* to 1000, specifying the batch size for mini-batch optimization. The Replay Buffer, which stores the newest samples after predictions are made, has a *max_length* of 1000, indicating the maximum number of samples it can store. A larger buffer provides more historical data for drift adaptation but also increases memory usage. The Drift Detection mechanism is configured with a threshold of 0.15, which means that performance changes exceeding this threshold of 15 % indicates a concept drift in the data stream. For instance, if the initial accuracy of the IDS is 90% and after processing several incremental batches of network traffic, the accuracy drops to 70% then this represents a 20% decline, which surpasses the defined threshold. This triggers the detection of a concept drift, prompting the model to update itself accordingly.

Lastly, Output Refinement includes a *sup_weight* parameter, which is the weight assigned to the supervised model's prediction in the final decision, which is set to 0.5. These parameter settings are crucial in optimizing model performance and ensuring accurate anomaly detection while balancing computational efficiency. The manual hyperparameter tuning is performed by a similar approach as presented in the previous chapters. Threshold optimization is set at 0.65 for anomaly scores to balance false alarms and detection accuracy.

We have investigated the relative performance of the hybrid incremental learning model across the initial and eight increments of data streams, each comprising network data packets. In the initial data stream, sniffed from the network, there are 100000 samples, and consequently, each incremental dataset consists of an additional 50000 data samples. Table 5.2 presents the performance results of the proposed model on each of these incremental datasets (I_1 to I_8). It was observed that the model's accuracy and F1-score consistently improved with each increment of training data, increasing from 79.93% to 98.88% accuracy and 78.56% to 98.64% F1-score. Simultaneously, the FPR decreased from 52.61% to 14.21%, indicating that the model's reliability improved with additional data. The FPR analysis reveals that while false alarms were initially

Table 5.1: Various model parameter settings and their values used in the experiments

Method	**Parameters**	**Parameter Value**
Hoeffding Tree	grace_period	100
	split_confidence	1e-5
	tie_threshold	0.05
Isolation Forest	estimators	200
	contamination	0.1
KMeans	Number of clusters	7
	mini_batch_size	1000
Replay Buffer	max_length	1000
Drift Detection	threshold	0.15
Output Refinement	sup_weight	0.5

high due to limited training exposure, the incremental learning strategy steadily reduced false positives to 14.21%. This consistent downward trend highlights the effectiveness of incremental learning in refining decision boundaries and reducing false alarms over time. Furthermore, it indicates that the hybrid incremental framework not only improves detection accuracy but also achieves operational practicality by significantly reducing false alerts. This highlights the advantage of incremental learning, where exposure to a larger volume of network traffic enhances model generalization and detection capability.

Table 5.2: Performance analysis of the proposed HIL-IDS model (%)

Metric	**Initial**	$\mathbf{I}_1$	$\mathbf{I}_2$	$\mathbf{I}_3$	$\mathbf{I}_4$	$\mathbf{I}_5$	$\mathbf{I}_6$	$\mathbf{I}_7$	$\mathbf{I}_8$
Data Samples	100000	150000	200000	250000	300000	350000	400000	450000	500000
Accuracy	79.93	82.75	85.08	87.13	88.89	91.66	94.24	96.90	**98.88**
F1-score	78.56	82.46	84.60	86.41	88.40	91.42	92.23	96.17	**98.64**
FPR	52.61	46.83	37.51	31.78	28.15	23.45	20.84	18.79	**14.21**

Additionally, the proposed model demonstrated efficient computational performance. The incremental approach significantly reduced the need for retraining on the entire

dataset, thereby lowering computational overhead. In real-world applications, this means the model can adapt dynamically to new network traffic patterns without requiring costly full retraining.

An error analysis of the model showed that most false positives occurred during the initial training stages, primarily due to the limited number of data samples available for learning normal and anomalous behaviours. However, as more data was introduced, the model was able to distinguish between legitimate and malicious network activity with higher confidence. Unlike static models that struggle with evolving threats, the proposed system leverages incremental learning to continuously refine its decision boundaries, ensuring robustness against emerging cyber threats.

The False Negative Rate (FNR), which reflects the proportion of actual attacks missed by the system, is equally critical in assessing the robustness of an intrusion detection framework. Although this chapter primarily reports Accuracy, F1-score, and FPR, the observed improvements in detection accuracy of 98.88% and F1-score of 98.64% across increments imply that FNR was very low, especially in later stages. In the early stages, when the model had limited exposure to representative attack traffic, the likelihood of missed attacks was higher, but this diminished steadily as the incremental updates provided more balanced exposure to diverse attack patterns, mitigating the risk of undetected attacks alongside their significant reduction in false positives.

Fig. 5.3 illustrates how the prediction time per sample (in seconds) of the proposed HIL-IDS model changes as more network data samples are incrementally added for training. The x-axis represents the different increments of data, starting from the Initial dataset and progressing through I_1 to I_8, while the y-axis represents the prediction time per sample in seconds.

The trend in the graph shows a clear downward slope, indicating that the prediction time per sample decreases with each incremental dataset. Initially, the prediction time per sample is 0.04900 seconds, which significantly drops to 0.00450 seconds by the eighth increment (I_8), marking an 89.8% reduction in time. The model becomes more efficient as it learns from a growing amount of data, leading to faster inference times. This efficiency gain can be attributed to the model adapting and optimizing its decision-making process over time, resulting in reduced computational complexity for each new sample it processes.

The most substantial improvement occurs in the early increments, particularly from

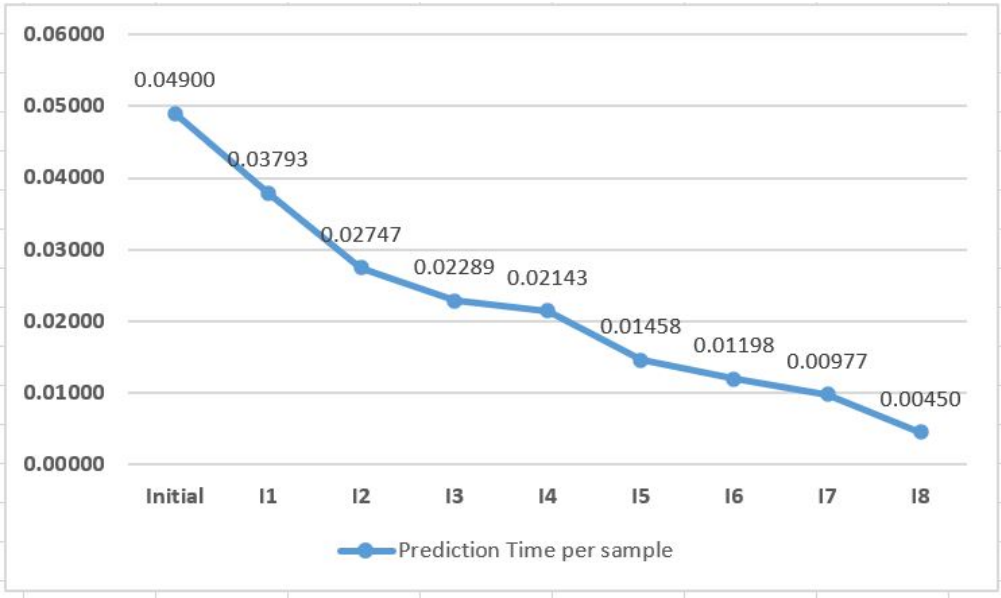

Fig. 5.3: Prediction time per sample (in sec) of the proposed model for each increment.

the Initial dataset to I_3, where the prediction time decreases rapidly, suggesting efficient feature learning and model optimization. After I_3, the rate of improvement becomes more gradual, indicating that the model has stabilized and further enhancements yield marginal gains. This consistent reduction in prediction time highlights the efficiency of the hybrid incremental learning approach, making the model more scalable and computationally efficient for real-time intrusion detection.

A comparative analysis of the proposed HIL-IDS model against state-of-the-art Machine Learning and Incremental Learning models in terms of accuracy, F1, and FPR is presented in Table 5.3.

Among the ML models, the Naïve Bayes classifier exhibits the lowest accuracy of 72.72%, an F1-score of 72.23%, and a high FPR of 48.31%. This indicates that while it performs well in certain cases, it is not well-suited for intrusion detection due to its assumption of feature independence and inability to effectively capture complex network attack patterns. On the other hand, Random forest showed an improved accuracy of 79.30%, an F1-score of 78.71%, and a reduced FPR of 39.73%.

For the incremental learning models, the Stochastic Gradient Descent (SGD) classifier, a linear model optimized using gradient descent, achieves an accuracy of 86.42%, an F1-score of 85.84%, and an FPR of 33.26%. This indicates that SGD performs bet-

Table 5.3: Comparative analysis of HIL-IDS with state-of-the-art ML and IL models (in %)

Model	Accuracy	F1-score	FPR
Naïve Bayes	72.72	72.23	48.31
Random Forest	79.30	78.71	39.73
Stochastic Gradient Descent	86.42	85.84	33.26
Online SVM	89.84	89.09	26.63
Hoeffding Tree	93.20	92.08	22.14
Proposed HIL-IDS	**98.88**	**98.64**	**14.21**

ter than Naïve Bayes and Random Forest by effectively learning from large-scale data, yet its false positive rate remains relatively high. The Online SVM further improves performance, reaching an accuracy of 89.84%, an F1-score of 89.09%, and an FPR of 26.63%. The reduced FPR indicates that SVM effectively distinguishes between normal and malicious traffic, although it may require significant computational resources.

The Hoeffding Tree, a widely used incremental learning model, demonstrates strong performance with an accuracy of 93.20%, an F1-score of 92.08%, and a further reduced FPR of 22.14%. This suggests that incremental learning techniques help adapt the model to evolving network threats while maintaining efficiency. However, despite its high accuracy, the Hoeffding Tree does not outperform the proposed model.

The proposed HIL-IDS achieved the most optimal results among all models, with an accuracy of 98.88%, an F1-score of 98.64%, and the lowest FPR of 14.21%. These results demonstrate the effectiveness of the hybrid incremental learning approach, which integrates both supervised and unsupervised learning to improve detection accuracy while minimizing false positives. The significant performance gain indicates that HIL-IDS efficiently learns from evolving data streams, adapts to new attack patterns, and reduces misclassification rates.

Overall, the results in Table 5.3 clearly highlight that traditional ML models such as Naïve Bayes, Random Forest, and SGD struggle to achieve high accuracy and suffer from a high false positive rate. Incremental learning models like Hoeffding Tree

and Online SVM improve performance, but they are still outperformed by the HIL-IDS framework, which effectively leverages incremental updates and hybrid learning strategies to deliver superior intrusion detection capabilities.

5.3.1 Statistical Significance

To validate whether the proposed HIL-IDS significantly outperforms the compared models, we applied the Friedman test across three evaluation metrics reported in Table 5.3, accuracy, F1-score, and FPR for six models. To maintain consistency across metrics and as higher values indicate better performance for accuracy and F1, while lower values are better for FPR, we used $(100 - FPR)$.

The Friedman statistical test was employed as it is a non-parametric statistical test widely used for comparing multiple algorithms across different evaluation scenarios. Each metric was treated as an independent evaluation dimension, as it captures distinct aspects of intrusion detection performance. We used $\alpha = 0.05$ (95% confidence). Using the F-statistic from Eqn. 3.11, the Friedman test yields the value of 12.11 with $p - value = 0.00278$, thus rejecting the null hypothesis (H_0) that all models perform equivalently. The average ranks of the models, as compared by the Friedman test, are presented in Table 5.4, demonstrating that HIL-IDS attains the best (lowest) average rank.

Table 5.4: Average Ranking of the Compared Models using the Friedman Test

Method	Average Ranking
HIL-IDS	1.00
Hoeffding Tree	2.00
Online SVM	3.00
SGD	4.00
Random Forest	5.00
Naïve Bayes	6.00

5.4 Chapter Summary

The Hybrid Incremental Learning-based Intrusion Detection System (HIL-IDS) is designed to detect anomalies in real-time network data streams, addressing the limitations of traditional intrusion detection systems that rely on static datasets and predefined attack signatures. The framework consists of four main phases: real-time network traffic ingestion, Apache Kafka pipeline integration, hybrid incremental learning, and output refinement. Network packets were captured using Scapy, which extracted 17 key features and applied pre-processing, normalization using Min-Max scaling, and dimensionality reduction with PCA to enhance model efficiency. The processed data was streamed employing Apache Kafka, ensuring fault tolerance and scalability. The core of the system is the hybrid incremental learning model that integrates a Hoeffding Tree classifier for supervised learning with Isolation Forest and KMeans clustering for unsupervised anomaly detection. The model assigns a combined confidence score to network packets, refining detection accuracy while minimizing false positives. A weighted anomaly score is used for classification, with a threshold of 0.65 optimized for balancing detection accuracy and false alarms. The model incorporates adaptive drift detection, enabling it to update itself when network patterns change over time. Experimental results, based on the CSE-CIC-IDS2018 dataset, demonstrate that HIL-IDS achieves a high accuracy of 98.88% and a low false positive rate of 14.21%, significantly outperforming traditional models such as Naïve Bayes, Random Forest, SGD, Online SVM, and Hoeffding Tree. The model also reduced the prediction time per sample, improving efficiency in real-time scenarios. Unlike static models, HIL-IDS continuously adapts to evolving attack patterns, making it a highly effective solution for modern cybersecurity challenges by providing scalable, real-time anomaly detection with high detection accuracy and minimal computational overhead. A Friedman test-based statistical analysis confirmed that the HIL-IDS model outperforms other incremental learning and ML models compared. Trained on the CSE-CIC-IDS2018 dataset and tested in a simulated real-time streaming setup, provided consistency and ensured rigorous benchmarking. While this establishes a reliable baseline, further validation on other datasets can broaden generalizability in future. Similarly, the current analysis addressed the attack classes present in the chosen dataset. Extending the study to include novel and emerging attack patterns will provide additional insights into robustness against previously unseen threats.

Chapter 6

Conclusion, Future Work, and Social Impact

In this thesis, we developed effective intrusion detection systems, which are crucial in fighting against evolving cyber threats. This chapter summarizes the key findings of this research, highlighting the contributions of the proposed work. Additionally, it explores potential directions for future research and discusses the broader social impact of this study on society.

6.1 Conclusion

In this research, we developed a novel technique for feature selection in an anomalous network environment, using the Dynamic Mutual Information-based Genetic Algorithm (DMI-GA). To compute the fitness value of the features, we introduce a novel fitness function based on an adaptive trade-off parameter. A comparative analysis of each model, consisting of the combination of ML with and without feature selection, has been statistically evaluated using two cross-validation strategies. The experimental results demonstrated that the detection accuracy of machine learning models improves significantly with the use of feature selection methods. Additionally, it was observed that among all the feature selection strategies employed with the ML model, DMI-GA consistently yielded the highest results across all combinational detection models. Specifically, Random Forest demonstrated superior performance compared to other ML

models when paired with the DMI-GA feature selection method, achieving the highest detection accuracy of 99.91% and an F1-score of 98.10%.

Furthermore, we designed and developed a robust intrusion detection system, evaluated on the up-to-date CSE-CIC-IDS2018 dataset. In the initial phase, we performed dimensionality reduction to enhance the model's performance. Since attack instances are often much rarer than benign traffic, this class imbalance can hinder the model's ability to accurately detect threats. Balanced datasets improve training by enabling the model to learn distinct features of both normal and malicious traffic. Hence, handling this imbalance is crucial for optimal performance. In this study, SMOTE is applied to address the class imbalance issue. After classifying network traffic as benign or malicious, specific attack types were identified within the attack class using the proposed modified Picture Fuzzy Clustering ($mP_{ic}FC$). A performance comparison of the proposed fuzzy clustering approach with state-of-the-art clustering techniques, including K-means, FCM, IFCM, and Picture Fuzzy Clustering, revealed that $mP_{ic}FC$ achieved the highest accuracy of 89.98%.

In today's digital era, securing networks against evolving cyber threats is crucial. Thus, in the final phase of our research, we developed HIL-IDS, a real-time intrusion detection system that leverages a hybrid incremental learning model to effectively identify network anomalies. This model combines the supervised Hoeffding Tree with an ensemble of unsupervised Isolation Forest and K-Means clustering. Real-time network traffic is captured by a sniffer, then preprocessed, feature-engineered, and streamed via the Apache Kafka pipeline to the hybrid model. If concept drift is detected, the model is updated, and predictions from subsequent increments are recalculated and verified. This iterative process continues until the final increment is used to classify incoming network packets as benign or anomalous. Before deployment in the real-time pipeline, the incremental learning model is pre-trained on the publicly available CSE-CIC-IDS2018 intrusion detection dataset. Experimental results show that as the number of ingested network samples increases, the model's performance improves, and the prediction time per sample decreases significantly.

6.2 Future Work

The methods developed in this thesis represent a significant advancement in the field of network security, particularly with intrusion detection. However, as with any research endeavor, there remain numerous opportunities for further exploration, enhancement, and deployment. The following are several key directions for future work:

- While the current methods demonstrate robust performance, the system's effectiveness could be tested and evaluated against emerging attack types to ensure its adaptability to evolving threats. Additionally, deploying and assessing the system in a real-world production environment is essential. For practical deployment, the system would be positioned behind the firewall to monitor both inbound and outbound network traffic. This setup would enable the system to analyze network activity, detect potential malicious attempts, and interact directly with the firewall, allowing it to take immediate action to block detected attacks.

- Future study will aim to enhance the IDS resilience against adversarial attacks, where attackers manipulate network traffic or features to evade detection. This will involve adversarial training, feature hardening, and ensemble learning to improve robustness.

- Building on the current work, future research will extend the scope to incorporate Explainable AI (XAI) techniques to provide insights into IDS decision-making, increasing trust and adoption in critical systems.

- Although Machine Learning methods have demonstrated significant results for intrusion detection, Deep Learning methods can offer additional capabilities that can further enhance detection performance, by automatically learning complex patterns from large volumes of network traffic data.

By addressing these areas, future research will not only enhance the effectiveness and robustness of intrusion detection systems but also ensure their adaptability to emerging threats and real-world environments. These advancements will ultimately lead to the development of more scalable and secure IDS solutions, better equipped to protect networks from evolving cyber threats in various applications.

6.3 Social Impact

The increasing reliance on digital infrastructure has amplified cybersecurity threats, making network security a crucial aspect of sustainable development. Effective intrusion detection systems play a vital role in safeguarding sensitive information, preventing financial losses, and maintaining national security. The development of a more accurate and efficient anomaly-based IDS has far-reaching implications for individuals, businesses, and government organizations. By enhancing network security, this research contributes to building a safer digital ecosystem and reducing the risks associated with cybercrime. Furthermore, it underscores the importance of integrating advanced machine learning techniques into cybersecurity frameworks to address emerging challenges. Machine learning enhances intrusion detection by enabling adaptive learning, where models continuously update themselves to detect evolving attack patterns.

In this context, the development of an advanced intrusion detection system also aligns with multiple United Nations Sustainable Development Goals (SDGs) by strengthening cybersecurity, protecting critical systems, and ensuring the confidentiality and integrity of digital operations. By effectively identifying and mitigating cyber threats, intrusion detection systems help prevent data breaches, safeguard sensitive information, and support the stability of essential services, as follows:

- **SDG 8 (*Decent Work and Economic Growth*):** Cyberattacks can cause significant financial losses to businesses, disrupt economic activities, and impact employment. The proposed efficient anomaly-based intrusion detection system would help businesses prevent cyber threats, ensure economic stability, and foster a secure work environment.

- **SDG 9 (*Industry, Innovation, and Infrastructure*):** Anomaly-based NIDS ensure robust network security measures, mitigate security risks, and promote innovation and sustainable industrial growth and technological advancements.

- **SDG 11 (*Sustainable Cities and Communities*):** As smart cities increasingly rely on interconnected digital systems, the IDS framework safeguards urban digital infrastructure by continuously monitoring network traffic and detecting potential cyber threats before they disrupt essential services.

- **SDG 16** ***(Peace, Justice, and Strong Institutions)*****:** By detecting and mitigating cyber threats in real time, the NIDS framework helps detect data breaches and cyberattacks on critical systems, strengthen digital governance, protect sensitive citizen data, and minimize cybercrime, contributing to a more secure and just society.

An anomaly-based network intrusion detection system strengthens cybersecurity resilience, enhances digital trust, and supports the sustainable growth of secure digital infrastructure.

References

[1] S.-W. Lee, H. M. Sidqi, M. Mohammadi, S. Rashidi, A. M. Rahmani, M. Masdari, and M. Hosseinzadeh, "Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review," *Journal of Network and Computer Applications*, vol. 187, no. 103111, pp. 1–22, 2021.

[2] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, 1987.

[3] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019.

[4] S. Seniaray and R. Jindal, "Machine learning-based network intrusion detection system," in *Computer Networks and Inventive Communication Technologies*, S. Smys, R. Bestak, R. Palanisamy, and I. Kotuliak, Eds. Singapore: Springer Nature Singapore, 2022, pp. 175–187.

[5] J. P. Anderson, "Computer security threat monitoring and surveillance," *Technical Report, James P. Anderson Company*, 1980.

[6] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE communications surveys & tutorials*, vol. 21, no. 1, pp. 686–728, 2018.

[7] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. Khan, "Performance anal-

ysis of machine learning algorithms in intrusion detection system: A review," *Procedia Computer Science*, vol. 171, pp. 1251–1260, 2020.

[8] A. Verma and V. Ranga, "Machine learning based intrusion detection systems for iot applications," *Wireless Personal Communications*, vol. 111, no. 4, pp. 2287–2310, 2020.

[9] R. Mitchell and I.-R. Chen, "A survey of intrusion detection in wireless network applications," *Computer Communications*, vol. 42, pp. 1–23, 2014.

[10] H. I. Ahmed, N. A. Elfeshawy, S. F. Elzoghdy, H. S. El-sayed, and O. S. Faragallah, "A neural network-based learning algorithm for intrusion detection systems," *Wireless Personal Communications*, vol. 97, pp. 3097—3112, 2017.

[11] J. Song, H. Takakura, and Y. Okabe, "Description of kyoto university benchmark data," *Available at link: http://www. takakura. com/Kyoto_data/BenchmarkData-Description-v5. pdf [Accessed on 15 March 2016]*, 2006.

[12] D. D. Protić, "Review of kdd cup '99, nsl-kdd and kyoto 2006+ datasets," *Vojnotehnički glasnik/Military Technical Courier*, vol. 66, no. 3, pp. 580–596, 2018.

[13] A. Thakkar and R. Lohiya, "A review of the advancement in intrusion detection datasets," *Procedia Computer Science*, vol. 167, pp. 636–645, 2020.

[14] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *computers & security*, vol. 31, no. 3, pp. 357–374, 2012.

[15] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani *et al.*, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, no. 2018, pp. 108–116, 2018.

[16] Canadian Institute for Cybersecurity, "CSE-CIC-IDS2018 Dataset," Dataset, 2018. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html

[17] A. Alenazi, I. Traore, K. Ganame, and I. Woungang, "Holistic model for http botnet detection based on dns traffic analysis," in *Intelligent, Secure, and Depend-*

able Systems in Distributed and Cloud Environments, I. Traore, I. Woungang, and A. Awad, Eds. Cham: Springer International Publishing, 2017, pp. 1–18.

[18] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," in *2019 international carnahan conference on security technology (ICCST)*. IEEE, 2019, pp. 1–8.

[19] J. Berkson, "Application of the logistic function to bio-assay," *Journal of the American Statistical Association*, vol. 39, no. 227, pp. 357–365, 1944.

[20] J. N. Morgan and J. A. Sonquist, "Problems in the analysis of survey data, and a proposal," *Journal of the American statistical association*, vol. 58, no. 302, pp. 415–434, 1963.

[21] W.-Y. Loh, "Fifty years of classification and regression trees," *International Statistical Review*, vol. 82, no. 3, pp. 329–348, 2014.

[22] T. Bayes, "Naive bayes classifier," *Article Sources and Contributors*, pp. 1–9, 1968.

[23] B. W. Silverman and M. C. Jones, "E. fix and jl hodges (1951): An important contribution to nonparametric discriminant analysis and density estimation: Commentary on fix and hodges (1951)," *International Statistical Review/Revue Internationale de Statistique*, vol. 57, no. 3, pp. 233–238, 1989.

[24] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[25] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[26] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.

[27] T. Evgeniou and M. Pontil, "Support vector machines: Theory and applications," in *Advanced course on artificial intelligence*. Springer, 1999, pp. 249–257.

[28] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[29] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection," *IEEE access*, vol. 6, pp. 33 789–33 795, 2018.

[30] O. Y. Al-Jarrah, Y. Al-Hammdi, P. D. Yoo, S. Muhaidat, and M. Al-Qutayri, "Semi-supervised multi-layered clustering model for intrusion detection," *Digital Communications and Networks*, vol. 4, no. 4, pp. 277–286, 2018.

[31] P. Tao, Z. Sun, and Z. Sun, "An improved intrusion detection algorithm based on ga and svm," *Ieee Access*, vol. 6, pp. 13 624–13 631, 2018.

[32] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the unsw-nb15 dataset," *Journal of Big Data*, vol. 7, no. 1, pp. 1–20, 2020.

[33] S. Biswas, "Intrusion detection using machine learning: A comparison study," *International Journal of Pure and Applied Mathematics*, vol. 118, pp. 101–114, 2018.

[34] G. Kocher and D. G. Kumar Ahuja, "Analysis of machine learning algorithms with feature selection for intrusion detection using unsw-nb15 dataset," *International Journal of Network Security Its Applications*, vol. 13, pp. 21–31, 2021.

[35] B. Uzun and S. Ballı, "A novel method for intrusion detection in computer networks by identifying multivariate outliers and relieff feature selection," *Neural Computing and Applications*, vol. 34, no. 20, pp. 1–16, 2022.

[36] M. B. Pranto, M. H. Ratul, M. Rahman, I. Jahan, and Z.-B. Zahir, "Performance of machine learning techniques in anomaly detection with basic feature selection strategy - a network intrusion detection system," *Journal of Advances in Information Technology*, vol. 13, pp. 36–4436, 2022.

[37] J. Ren, J. Guo, W. Qian, H. Yuan, X. Hao, and H. Jingjing, "Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms," *Security and communication networks*, vol. 2019, no. 1, p. 7130868, 2019.

[38] A. Thakkar and R. Lohiya, "Attack classification using feature selection techniques: a comparative study," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 1249–1266, 2021.

[39] Z. Halim, M. N. Yousaf, M. Waqas, M. Sulaiman, G. Abbas, M. Hussain, I. Ahmad, and M. Hanif, "An effective genetic algorithm-based feature selection method for intrusion detection systems," *Computers & Security*, vol. 110, p. 102448, 2021.

[40] B. Kaushik, R. Sharma, K. Dhama, A. Chadha, and S. Sharma, "Performance evaluation of learning models for intrusion detection system using feature selection," *Journal of Computer Virology and Hacking Techniques*, vol. 19, no. 4, pp. 529–548, 2023.

[41] M. Awad and S. Fraihat, "Recursive feature elimination with cross-validation with decision tree: Feature selection method for machine learning-based intrusion detection systems," *Journal of Sensor and Actuator Networks*, vol. 12, no. 5, 2023.

[42] A. V. Turukmane and R. Devendiran, "M-multisvm: An efficient feature selection assisted network intrusion detection system using machine learning," *Computers & Security*, vol. 137, 2024.

[43] Y. Akhiat, K. Touchanti, A. Zinedine, and M. Chahhou, "Ids-efs: Ensemble feature selection-based method for intrusion detection system," *Multimedia Tools and Applications*, vol. 83, no. 5, pp. 12 917–12 937, 2024.

[44] Q. R. S. Fitni and K. Ramli, "Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems," in *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*. Bali, Indonesia: IEEE, 2020, pp. 118–124.

[45] S. B. Mallampati and S. Hari, "Fusion of feature ranking methods for an effective intrusion detection system." *Computers, Materials & Continua*, vol. 76, no. 2, pp. 1721–1744, 2023.

[46] N. G. Pardeshi and D. V. Patil, "Applying gini importance and rfe methods for feature selection in shallow learning models for implementing effective intrusion detection system," in *International Conference on Applications of Machine Intelligence and Data Analytics (ICAMIDA 2022)*. Aurangabad, India: Atlantis Press, 2023, pp. 214–234.

[47] M. Verkerken, L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, "Towards model generalization for intrusion detection: Unsupervised machine learning techniques," *Journal of Network and Systems Management*, vol. 30, pp. 1–25, 2021.

[48] F. Salo, A. B. Nassif, and A. Essex, "Dimensionality reduction with ig-pca and ensemble classifier for network intrusion detection," *Computer networks*, vol. 148, pp. 164–175, 2019.

[49] G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the performance of machine learning-based idss on an imbalanced and up-to-date dataset," *IEEE access*, vol. 8, pp. 32 150–32 162, 2020.

[50] N. Oliveira, I. Praça, E. Maia, and O. Sousa, "Intelligent cyber attack detection and classification for network-based intrusion detection systems," *Applied Sciences*, vol. 11, no. 4, 2021.

[51] X. Larriva-Novo, C. Sánchez-Zas, V. A. Villagrá, M. Vega-Barbas, and D. Rivera, "An approach for the application of a dynamic multi-class classifier for network intrusion detection systems," *Electronics*, vol. 9, no. 11, 2020.

[52] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, 2019.

[53] T.-H. Chua and I. Salam, "Evaluation of machine learning algorithms in network-based intrusion detection system," *arXiv preprint arXiv:2203.05232*, 2022.

[54] L. Liu, P. Wang, J. Lin, and L. Liu, "Intrusion detection of imbalanced network traffic based on machine learning and deep learning," *IEEE access*, vol. 9, pp. 7550–7563, 2020.

[55] O. H. Abdulganiyu, T. A. Tchakoucht, Y. K. Saheed, and H. A. Ahmed, "Xidintfl-vae: Xgboost-based intrusion detection of imbalance network traffic via class-wise focal loss variational autoencoder," *The Journal of Supercomputing*, vol. 81, no. 1, pp. 1–38, 2024.

[56] B. Harish and S. A. Kumar, "Anomaly based intrusion detection using modified fuzzy clustering," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, no. 6, pp. 54–59, 2017.

[57] K. Samunnisa, G. S. V. Kumar, and K. Madhavi, "Intrusion detection system in distributed cloud computing: Hybrid clustering and classification methods," *Measurement: Sensors*, vol. 25, pp. 1–12, 2023.

[58] P. S. Bhattacharjee, A. K. M. Fujail, and S. A. Begum, "A comparison of intrusion detection by k-means and fuzzy c-means clustering algorithm over the nsl-kdd dataset," in *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*. Coimbatore, India: IEEE, 2017, pp. 1–6.

[59] Z. Rustam and N. P. A. A. Ariantari, "Comparison between support vector machine and fuzzy kernel c-means as classifiers for intrusion detection system using chi-square feature selection," in *AIP Conference Proceedings of the 3rd International Symposium on Current Progress in Mathematics and Sciences 2017 (ISCPMS2017)*, vol. 2023, no. 1. Bali, Indonesia: AIP Publishing, 2018, pp. 1–7.

[60] G.-Y. Shin, D.-W. Kim, S.-S. Kim, and M.-M. Han, "Unknown attack detection: Combining relabeling and hybrid intrusion detection." *Computers, Materials & Continua*, vol. 68, no. 3, 2021.

[61] S. M. Kasongo, "A deep learning technique for intrusion detection system using a recurrent neural networks based framework," *Computer Communications*, vol. 199, pp. 113–125, 2023.

[62] R. B. Said, Z. Sabir, and I. Askerzade, "Cnn-bilstm: A hybrid deep learning approach for network intrusion detection system in software-defined networking with hybrid feature selection," *IEEE Access*, vol. 11, pp. 138 732–138 747, 2023.

[63] M. Sajid, K. R. Malik, A. Almogren, T. S. Malik, A. H. Khan, J. Tanveer, and A. U. Rehman, "Enhancing intrusion detection: a hybrid machine and deep learning approach," *Journal of Cloud Computing*, vol. 13, no. 1, p. 123, 2024.

[64] Y.-C. Wang, Y.-C. Houng, H.-X. Chen, and S.-M. Tseng, "Network anomaly intrusion detection based on deep learning approach," *Sensors*, vol. 23, no. 4, p. 2171, 2023.

[65] M. Farhan, H. Waheed ud din, S. Ullah, M. S. Hussain, M. A. Khan, T. Mazhar, U. F. Khattak, and I. H. Jaghdam, "Network-based intrusion detection using deep learning technique," *Scientific Reports*, vol. 15, no. 1, p. 25550, 2025.

[66] M. A. Alsharaiah, A. A. Abu-Shareha, M. Abualhaj, L. H. Baniata, A. Al-saaidah, Q. M. Kharma, and M. M. Al-Zyoud, "An innovative network intrusion detection system (nids): Hierarchical deep learning model based on unsw-nb15 dataset." *International Journal of Data & Network Science*, vol. 8, no. 2, 2024.

[67] F. S. Alrayes, M. Zakariah, S. U. Amin, Z. I. Khan, and J. S. Alqurni, "Network security enhanced with deep neural network-based intrusion detection system." *Computers, Materials & Continua*, vol. 80, no. 1, 2024.

[68] S. Elsayed, K. Mohamed, and M. A. Madkour, "A comparative study of using deep learning algorithms in network intrusion detection," *IEEE Access*, vol. 12, pp. 58 851–58 870, 2024.

[69] S. Hore, J. Ghadermazi, A. Shah, and N. D. Bastian, "A sequential deep learning framework for a robust and resilient network intrusion detection system," *Computers & Security*, vol. 144, p. 103928, 2024.

[70] I. M. Sayem, M. I. Sayed, S. Saha, and A. Haque, "Enids: A deep learning-based ensemble framework for network intrusion detection systems," *IEEE transactions on network and service management*, vol. 21, no. 5, pp. 5809–5825, 2024.

[71] A. Thaljaoui, "Intelligent network intrusion detection system using optimized deep cnn-lstm with unsw-nb15," *International Journal of Information Technology*, pp. 1–17, 2025.

[72] N. O. Aljehane, H. A. Mengash, M. M. Eltahir, F. A. Alotaibi, S. S. Aljameel, A. Yafoz, R. Alsini, and M. Assiri, "Golden jackal optimization algorithm with deep learning assisted intrusion detection system for network security," *Alexandria Engineering Journal*, vol. 86, pp. 415–424, 2024.

[73] Y. Dong, R. Wang, and J. He, "Real-time network intrusion detection system based on deep learning," in *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*. Beijing, China: IEEE, 2019, pp. 1–4.

[74] F. A. P. Kuswara, H. H. Nuha, and V. Suryani, "Intrusion detection system using incremental learning method," in *2023 11th International Conference on Information and Communication Technology (ICoICT)*. Melaka, Malaysia: IEEE, 2023, pp. 588–593.

[75] S.-T. Wang and S.-S. Sun, "An incremental learning model for network intrusion detection systems," in *2024 10th International Conference on Applied System Innovation (ICASI)*. Kyoto, Japan: IEEE, 2024, pp. 362–364.

[76] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," *arXiv preprint arXiv:1802.09089*, 2018.

[77] H. Zhang, S. Dai, Y. Li, and W. Zhang, "Real-time distributed-random-forest-based network intrusion detection system using apache spark," in *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*, Orlando, FL, USA, 2018, pp. 1–7.

[78] M. T. Tun, D. E. Nyaung, and M. P. Phyu, "Performance evaluation of intrusion detection streaming transactions using apache kafka and spark streaming," in *2019 International Conference on Advanced Information Technologies (ICAIT)*, Yangon, Myanmar, 2019, pp. 25–30.

[79] S. M. H. Bamakan, H. Wang, T. Yingjie, and Y. Shi, "An effective intrusion detection framework based on mclp/svm optimized by time-varying chaos particle swarm optimization," *Neurocomputing*, vol. 199, pp. 90–102, 2016.

[80] X.-s. Gan, J.-s. Duanmu, J.-f. Wang, and W. Cong, "Anomaly intrusion detection based on pls feature extraction and core vector machine," *Knowledge-Based Systems*, vol. 40, pp. 1–6, 2013.

[81] B. Luo and J. Xia, "A novel intrusion detection system based on feature generation with visualization strategy," *Expert Systems with Applications*, vol. 41, no. 9, pp. 4139–4147, 2014.

[82] U. Banerjee, A. Vashishtha, and M. Saxena, "Evaluation of the capabilities of wireshark as a tool for intrusion detection," *International Journal of computer applications*, vol. 6, no. 7, pp. 1–5, 2010.

[83] S. Pavithirakini, D. Bandara, C. Gunawardhana, K. Perera, B. Abeyrathne, and D. Dhammearatchi, "Improve the capabilities of wireshark as a tool for intrusion detection in dos attacks," *International Journal of Scientific and Research Publications*, vol. 6, no. 4, pp. 378–384, 2016.

[84] Amrita and Ahmed, "A study of feature selection methods in intrusion detection system: A survey," *International Journal of Computer Science Engineering and Information Technology Research*, vol. 2, no. 3, pp. 1–25, 2012.

[85] H. T. Nguyen, S. Petrović, and K. Franke, "A comparison of feature-selection methods for intrusion detection," in *Kotenko, I., Skormin, V. (eds) Computer Network Security. MMM-ACNS 2010. Lecture Notes in Computer Science*, vol. 6258. Berlin, Heidelberg: Springer, 2010, pp. 242–255.

[86] M. A. Hall and L. A. Smith, "Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper," in *Twelfth International FLAIRS conference*, Florida, USA, 1999, pp. 235–239.

[87] E. Ghiselli, *Theory of Psychological Measurement*, ser. McGraw-Hill series in psychology. McGraw-Hill, 1964.

[88] R. A. Ghazy, E.-S. M. El-Rabaie, M. I. Dessouky, N. A. El-Fishawy, and F. E. A. El-Samie, "Feature selection ranking and subset-based techniques with different classifiers for intrusion detection," *Wireless Personal Communications*, vol. 111, pp. 375—393, 2020.

[89] J. R. Quinlan, "Learning decision tree classifiers," *ACM Computing Surveys (CSUR)*, vol. 28, no. 1, pp. 71–72, 1996.

[90] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning*, vol. 46, pp. 389–422, 2002.

[91] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–73, 1992.

[92] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on neural networks*, vol. 5, no. 4, pp. 537–550, 1994.

[93] U. F. Siddiqi, S. M. Sait, and O. Kaynak, "Genetic algorithm for the mutual information-based feature selection in univariate time series data," *IEEE Access*, vol. 8, pp. 9597–9609, 2020.

[94] O. Elzeki, M. Alrahmawy, and S. Elmougy, "A new hybrid genetic and information gain algorithm for imputing missing values in cancer genes datasets," *International Journal of Intelligent Systems and Applications*, vol. 10, no. 12, pp. 20–33, 2019.

[95] A. M. Al Tobi and I. Duncan, "Improving intrusion detection model prediction by threshold adaptation," *Information*, vol. 10, no. 5, 159, pp. 1–42, 2019.

[96] Y. Yang and G. I. Webb, "A comparative study of discretization methods for naive-bayes classifiers," in *Proceedings of PKAW 2002, The 2002 Pacific Rim Knowledge Acquisition Workshop*, Tokyo, Japan, 2002, pp. 159–173.

[97] H. A. Salman, A. Kalakech, and A. Steiti, "Random forest algorithm overview," *Babylonian Journal of Machine Learning*, vol. 2024, pp. 69–79, 2024.

[98] H. Xue, S. Chen, and Q. Yang, "Structural regularized support vector machine: a framework for structural large margin classifier," *IEEE Transactions on Neural Networks*, vol. 22, no. 4, pp. 573–587, 2011.

[99] S. Das, S. Saha, A. T. Priyoti, E. K. Roy, F. T. Sheldon, A. Haque, and S. Shiva, "Network intrusion detection and comparative analysis using ensemble machine learning and feature selection," *IEEE transactions on network and service management*, vol. 19, no. 4, pp. 4821–4833, 2021.

[100] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Scientific Reports*, vol. 14, no. 6086, 2024.

[101] T. Gupta, R. Jindal, and I. Sreedevi, "Empirical review of various thermography-based computer-aided diagnostic systems for multiple diseases," *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 3, pp. 1–33, 2023.

[102] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.

[103] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, pp. 1–30, 2006.

[104] F. W. L. W. P. A. Stolfo, Salvatore and P. Chan, "KDD Cup 1999 Data," UCI Machine Learning Repository, 1999, DOI: https://doi.org/10.24432/C51C7N.

[105] T. Eldos, M. K. Siddiqui, and A. Kanan, "On the kdd'99 dataset: Statistical analysis for feature selection," *Journal of Data Mining and Knowledge Discovery*, vol. 3, no. 3, p. 88, 2012.

[106] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[107] I. Ullah and Q. H. Mahmoud, "A scheme for generating a dataset for anomalous activity detection in iot networks," in *Canadian conference on artificial intelligence*. Springer, 2020, pp. 508–520.

[108] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment," *Sensors*, vol. 23, no. 13, p. 5941, 2023.

[109] V. A. Phan, J. Jerabek, and L. Malina, "Comparison of multiple feature selection techniques for machine learning-based detection of iot attacks," in *Proceedings of the 19th International Conference on Availability, Reliability and Security*, 2024, pp. 1–10.

[110] M. A. Khan, N. Iqbal, H. Jamil, D.-H. Kim *et al.*, "An optimized ensemble prediction model using automl based on soft voting classifier for network intrusion detection," *Journal of Network and Computer Applications*, vol. 212, p. 103560, 2023.

[111] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[112] A. S. Barkah, S. R. Selamat, Z. Z. Abidin, and R. Wahyudi, "Impact of data balancing and feature selection on machine learning-based network intrusion detection," *JOIV: International Journal on Informatics Visualization*, vol. 7, no. 1, pp. 241–248, 2023.

[113] H. Shah, J. Undercoffer, and A. Joshi, "Fuzzy clustering for intrusion detection," in *The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ'03.*, vol. 2. St Louis, MO, USA: IEEE, 2003, pp. 1274–1278.

[114] W. Ren, J. Cao, and X. Wu, "Application of network intrusion detection based on fuzzy c-means clustering algorithm," in *2009 Third International Symposium on Intelligent Information Technology Application*, vol. 3. Nanchang, China: IEEE, 2009, pp. 19–22.

[115] P. H. Thong and L. H. Son, "Picture fuzzy clustering: a new computational intelligence method," *Soft computing*, vol. 20, no. 9, pp. 3549–3562, 2016.

[116] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[117] K. T. Atanassov, *Intuitionistic fuzzy sets*. Springer: Physica-Verlag HD, 1999, vol. 35.

[118] J. C. Bezdek, "Objective function clustering," *Pattern recognition with fuzzy objective function algorithms*, pp. 43–93, 1981.

[119] R. R. Yager, "On the measure of fuzziness and negation part i: membership in the unit interval," *International Journal of General Systems*, vol. 5, no. 4, pp. 221–229, 1979.

[120] R. Yager, "On the measure of fuzziness and negation. ii. lattices," *Information and control*, vol. 44, no. 3, pp. 236–260, 1980.

[121] B. C. Cuong, "Picture fuzzy sets," *Journal of Computer Science and Cybernetics*, vol. 30, no. 4, pp. 409–409, 2014.

[122] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[123] D. Kumar, R. Agrawal, and H. Verma, "Kernel intuitionistic fuzzy entropy clustering for mri image segmentation," *Soft Computing*, vol. 24, pp. 4003–4026, 2020.

[124] Z. Xu and J. Wu, "Intuitionistic fuzzy c-means clustering algorithms," *Journal of Systems Engineering and Electronics*, vol. 21, no. 4, pp. 580–590, 2010.

[125] A. Panwar, "A kernel based atanassov's intuitionistic fuzzy clustering for network forensics and intrusion detection," in *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)*. Las Vegas, NV, USA: IEEE, 2015, pp. 107–112.

[126] T. Chaira, "A novel intuitionistic fuzzy c means clustering algorithm and its application to medical images," *Applied soft computing*, vol. 11, no. 2, pp. 1711–1717, 2011.

[127] E. Szmidt and J. Kacprzyk, "Distances between intuitionistic fuzzy sets," *Fuzzy sets and systems*, vol. 114, no. 3, pp. 505–518, 2000.

[128] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on cse-cic-ids2018 big data," *Journal of Big Data*, vol. 7, pp. 1–19, 2020.

[129] S. Patro and K. K. Sahu, "Normalization: A preprocessing stage," *arXiv preprint arXiv:1503.06462*, 2015.

[130] Z. Liu *et al.*, "A method of svm with normalization in intrusion detection," *Procedia Environmental Sciences*, vol. 11, pp. 256–262, 2011.

[131] M. A. Umar, Z. Chen, K. Shuaib, and Y. Liu, "Effects of feature selection and normalization on network intrusion detection," *Authorea Preprints*, 2024.

[132] S. Songma, T. Sathuphan, and T. Pamutha, "Optimizing intrusion detection systems in three phases on the cse-cic-ids-2018 dataset," *Computers*, vol. 12, no. 12, 245, pp. 1–20, 2023.

[133] N. Patel, B. Mehtre, and R. Wankar, "A computationally efficient dimensionality reduction and attack classification approach for network intrusion detection," *International Journal of Information Security*, vol. 23, pp. 2457–2487, 2024.

[134] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of educational psychology*, vol. 24, no. 6, pp. 417–441, 1933.

[135] V. Venkatachalam and S. Selvan, "Performance comparison of intrusion detection system classifiers using various feature reduction techniques," *International journal of simulation*, vol. 9, no. 1, pp. 30–39, 2008.

[136] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[137] Z. Elkhadir and B. Mohammed, "A cyber network attack detection based on gm median nearest neighbors lda," *computers & security*, vol. 86, pp. 63–74, 2019.

[138] J. C. Bezdek, *Fuzzy-Mathematics in Pattern Classification*. Cornell University, 1973.

[139] G. Wang, J. Hao, J. Ma, and L. Huang, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering," *Expert systems with applications*, vol. 37, no. 9, pp. 6225–6232, 2010.

[140] S. Seniaray and R. Jindal, "Performance analysis of anomaly-based network intrusion detection using feature selection and machine learning techniques," *Wireless Personal Communications*, vol. 138, pp. 2321–2351, 2024.

[141] R. Sampat and S. Sonawani, "A survey of fuzzy clustering techniques for intrusion detection system," *Int J Eng Res Technol*, vol. 3, no. 1, pp. 2188–2192, 2014.

[142] C.-M. Yang, Y. Liu, Y.-T. Wang, Y.-P. Li, W.-H. Hou, S. Duan, and J.-Q. Wang, "A novel adaptive kernel picture fuzzy c-means clustering algorithm based on grey wolf optimizer algorithm," *Symmetry*, vol. 14, no. 7, 1442, pp. 1–22, 2022.

[143] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "Cluster validity methods: part i," *ACM Sigmod Record*, vol. 31, no. 2, pp. 40–45, 2002.

[144] R. K. Verma, R. Tiwari, and P. S. Thakur, "Partition coefficient and partition entropy in fuzzy c means clustering," *Journal of Scientific Research and Reports*, vol. 29, no. 12, pp. 1–6, 2023.

[145] W. Wang and Y. Zhang, "On fuzzy cluster validity indices," *Fuzzy sets and systems*, vol. 158, no. 19, pp. 2095–2117, 2007.

[146] K. R. Žalik, "Cluster validity index for estimation of fuzzy clusters of different sizes and densities," *Pattern Recognition*, vol. 43, no. 10, pp. 3374–3390, 2010.

[147] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 08, pp. 841–847, 1991.

[148] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Transactions on Fuzzy systems*, vol. 3, no. 3, pp. 370–379, 1995.

[149] T. Gupta, R. Jindal, and I. Sreedevi, "Empirical review of various thermography-based computer-aided diagnostic systems for multiple diseases," *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 3, pp. 1–33, 2023.

[150] A. S. Sha'ari and Z. Abdullah, "A comparative study between machine learning and deep learning algorithm for network intrusion detection," *Journal of Soft Computing and Data Mining*, vol. 3, no. 2, pp. 43–51, 2022.

[151] Y. Zhang, H. Zhang, and B. Zhang, "An effective ensemble automatic feature selection method for network intrusion detection," *Information*, vol. 13, no. 7, 314, pp. 1–15, 2022.

[152] A. Elhanashi, K. Gasmi, A. Begni, P. Dini, Q. Zheng, and S. Saponara, "Machine learning techniques for anomaly-based detection system on cse-cic-ids2018 dataset," in *International Conference on Applications in Electronics Pervading Industry, Environment and Society*. Genoa, Italy: Springer, 2023, pp. 131–140.

[153] S. Brahmanand, N. D. Lal, D. Sahana, G. Nijguna, and P. Nayak, "A systematic approach of analysing network traffic using packet sniffing with scapy framework," in *Computer Networks and Inventive Communication Technologies: Proceedings of Fourth ICCNCT 2021*, S. Smys, R. Bestak, R. Palanisamy, and I. Kotuliak, Eds. Singapore: Springer Nature Singapore, 2022, pp. 811–820.

[154] K. M. M. Thein, "Apache kafka: Next generation distributed messaging system," *International Journal of Scientific Engineering and Technology Research*, vol. 3, no. 47, pp. 9478–9483, 2014.

[155] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, Boston, MA, 2000, pp. 71–80.

[156] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. Pisa, Italy: IEEE, 2008, pp. 413–422.

[157] M. A. Umar, Z. Chen, K. Shuaib, and Y. Liu, "Effects of feature selection and normalization on network intrusion detection," *Data Science and Management*, vol. 8, no. 1, pp. 23–39, 2025.

[158] M. J. Sax, "Apache kafka," in *Encyclopedia of Big Data Technologies*, S. Sakr and A. Zomaya, Eds. Cham: Springer International Publishing, 2018, pp. 1–8.

[159] C. MS, S. BK, M. F. Beejady, R. BS *et al.*, "Kafka-based intrusion detection system," *Grenze International Journal of Engineering & Technology*, vol. 10, no. 2, p. 1472, 2024.

[160] A.-T. Costin, D. Zinca, and V. Dobrota, "A real-time streaming system for customized network traffic capture," *Sensors*, vol. 23, no. 14, 6467, pp. 1–17, 2023.

[161] T. P. Raptis and A. Passarella, "A survey on networked data streaming with apache kafka," *IEEE access*, vol. 11, pp. 85 333–85 350, 2023.

[162] V. Mehta and V. Sanghavi, "Comparative study of various decision tree methods for data stream mining," in *Third International Congress on Information and Communication Technology: ICICT 2018*. London, UK: Springer, 2019, pp. 371–379.

LIST OF PUBLICATIONS

SCI-INDEXED JOURNALS (Published/ Accepted)

- **Sumedha Seniaray** and Rajni Jindal,"Performance Analysis of Anomaly-Based Network Intrusion Detection Using Feature Selection and Machine Learning Techniques," *Wireless Personal Communications*, vol. 138, pp. 2321-2351, October 2024, DOI: 10.1007/s11277-024-11602-5. **Impact factor: 2.2** (***Published***)

- **Sumedha Seniaray** and Rajni Jindal, "Enhanced Anomaly-based Network Intrusion Detection Leveraging a Modified Picture Fuzzy Clustering Approach," *Cluster Computing*, vol. 28 (7), pp. 1-25, July 2025, DOI: 10.1007/s10586-024-04952-z. **Impact factor: 4.1** (***Published***)

SCOPUS-INDEXED CONFERENCES (Published/ Accepted)

- **Sumedha Seniaray** and Rajni Jindal, "Machine Learning-Based Network Intrusion Detection System," *Smys, S., Bestak, R., Palanisamy, R., Kotuliak, I. (eds) Computer Networks and Inventive Communication Technologies. Lecture Notes on Data Engineering and Communications Technologies*, vol. 75, Springer, Singapore, 2021, pp. 175-187, DOI: 10.1007/978-981-16-3728-5_13. **(*Published*)**

- **Sumedha Seniaray** and Rajni Jindal, "Real-time Network Intrusion Detection using Hybrid Incremental Learning Approach," *Innovative Computing and Communications (ICICC 2025), Lecture Notes in Networks and Systems,* vol. 1438. Springer, Singapore, 2025, DOI: 10.1007/978-981-96-7707-8_34. ***(Published)***

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahabad Daulatpur, Main Bawana Road, Delhi-110042, INDIA

PLAGIARISM VERIFICATION

Title of the Thesis: **Anomaly-based Network Intrusion Detection System using Machine Learning Algorithms**
Total Pages: **157 pages (Abstract onwards)**
Name of the Scholar: **Sumedha Seniaray**
Roll No.: 2K19/PHDCO/16
Department: **Computer Science & Engineering**
Supervisor: **Prof. Rajni Jindal, Department of Computer Science & Engineering**

This is to report that the above-mentioned thesis was scanned for similarity detection. Process and outcome is given below:
Software used: Turnitin
Similarity index: **5% (= 38% - 33% (Self-Plagiarism))**
Total Word Count: **45,794**

Candidate's Signature - Sumedha Seniaray

Prof. Rajni Jindal
(Supervisor)
(Department of Computer Science & Engineering)

Date: 08/10/2025
Place: New Delhi

Author Biography

Sumedha Seniaray
Assistant Professor,
Department of Applied Mathematics
Delhi Technological University, Delhi, India
Email: sumedhaseniaray@dtu.ac.in

Sumedha Seniaray received her M.Tech. (Computer Science & Engineering) degree from the Centre for Development of Advanced Computing (C-DAC), Noida, India. She has been serving as an Assistant Professor at Delhi Technological University (DTU), Delhi, India, since 2018. She is pursuing her Ph.D. in the Computer Science & Engineering Department at DTU. Her research interests include network security and cybersecurity.