INTRUSION DETECTION SYSTEM USING DEEP LEARNING

THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

MASTER OF TECHNOLOGY
in
ARTIFICIAL INTELLIGENCE

Submitted by

SANCHIT AGARWAL
(2K22/AFI/19)

Under the supervision of
Dr. Pawan Singh Mehra

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

MAY, 2024

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, **Sanchit Agarwal**, Roll No – **(2K21/AFI/19)** student of M.Tech (**Department of Computer Science Engineering**), hereby declare that the project Dissertation titled "**INTRUSION DETECTION SYSTEM USING DEEP LEARNING**" which is submitted by me to the **Department of Computer Science Engineering**, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma, Associateship, Fellowship or other similar title or recognition.

Place: Delhi **Sanchit Agarwal**

Date : 31.05.2024 **(2K22/AFI/19)**

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 my knowledge.

Signature of Supervisor **Signature of External Examiner**

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Thesis report titled **"INTRUSION DETECTION SYSTEM USING DEEP LEARNING"** which is submitted by Sanchit Agarwal, Roll No. 2K22/AFI/19, Department of Computer Science Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Date: 31.05.2024

Dr. Pawan Singh Mehra
SUPERVISOR

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I wish to express my sincerest gratitude to **Dr. Pawan Singh Mehra** for his continuous guidance and mentorship that he provided me during the project. He showed me the path to achieve my targets by explaining all the tasks to be done and explained to me the importance of this project as well as its industrial relevance. He was always ready to help me and clear my doubts regarding any hurdles in this project. Without his constant support and motivation, this project would not have been successful.

Place: Delhi **Sanchit Agarwal**

Date: 31.05.2024 **(2K22/AFI/19)**

Abstract

Intrusion detection systems should be powerful and reliable in the age of the Internet of Things to ensure the security and integrity of interconnecting devices. In this thesis, we employ deep learning augmentation techniques using Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) networks. We tested the performances of the models on three benchmark datasets: NSL-KDD, UNSW-NB15, and CICIDS 2017. Our focus was on the ability of the models to classify data into normal and attack classes. We show in this work that both models are highly efficacious, though with some variation in the performance metrics across different scenarios. In the case of data sets, the BiLSTM model outperformed the LSTM in most metrics, with accuracies of over 98% in all cases and an excellent result in the UNSW-NB15 dataset of over 99%. This comparative analysis not only allows us to know the potential of the LSTM and BiLSTM models in the domain of IoT IDS, but also their operational strengths and weaknesses across diverse attack scenarios, which could guide further research and practical implementations of deep learning-based IDS in the enhancement of IoT security.

Keywords-Machine Learning, Deep learning, LSTM, Bi-LSTM, Intrusion Detection.

TABLE OF CONTENTS

List of Tables

List of Figures

List of Abbreviations

ML - Machine Learning

DL - Deep Learning

AI - Artificial Intelligence

IoT - Internet of Things

IDS - Intrusion Detection System

SVM - Support Vector Machine

KNN - K-Nearest Neighbour

PCA - Principal Component Analysis

ANN - Artificial Neural Network

CNN - Convolutional Neural Network

RNN - Recurrent Neural Network

LSTM - Long-Short Term Memory

Bi-LSTM - Bi Directional Long-Short Term Memory

BERT - Bidirectional Encoder Representations from Transformers

CHAPTER 1
INTRODUCTION

1.1 Machine Learning

Machine learning is a part of artificial intelligence that has been contributing to breakthroughs in various fields and becoming increasingly prevalent. ML enables systems to learn and evolve without being explicitly programmed. It relies on the creation of algorithms allowing to interpret and learn from data but is not limited to them. One of the rapidly developing application areas of machine learning is that concerning information security. More specifically, this report regards intrusion detection as one of the areas to benefit from machine learning applications.

Intrusion detection system, or IDS, is an inspection tool that monitors the activities of a network or a system for malicious or policy-violating behaviours. IDS has been gradually shifting from a human-controlled and managed system to a stand-alone device. The man-controlled approach implies that IDS is set up by human operators defining the rules or using the signature-based method. The problem is that new rules for a vast influx of new cyber threats are too numerous and complex to be identified independently. This is the reason why machine learning in IDS is a phenomenon helping to identify patterns and threats that a human could not notice.

Machine learning models can be trained on large datasets with numerous types of normal and abnormal inputs and signals, for example, various forms of traffic, and attack vectors. Through the use of algorithms such as decision trees, support vector machines, or even more complex forms of neural networks, a machine learning model can effectively distinguish between benign and malicious forms of activity.

This is a clear advantage in fields such as the Internet of Things, where the numbers and types of entities and interactions are highly varied and not singular in form. This

means that normal and abnormal activity is not easily generalized and new forms of activity are constantly being discovered.

Additionally, in the context of the IoT, there is the problem of the rapidly changing perimeter of the system, which expands by one with each new device. Furthermore, machine learning makes it easier to adapt to new forms of existing threats. Existing solutions do not always adapt to new types of attacks, and they emerge almost daily. Therefore, machine learning can enable the production of IDS that will continue to adapt and update themselves as new vulnerabilities are discovered and exploited.

Finally, integrating machine learning in IDS allows for more proactive security. This is another way of saying that with machine learning solutions, it is easier to deviate from the update schedule based on known threats. Instead of just protecting against what is known, machine learning systems can proactively seek and eliminate potential vulnerabilities before they are exploited by attackers.

1.2 Categories of Machine Learning

Machine learning can be categorized into four types depending on the nature of the learning signal or feedback available to the system: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Each type can be applied to different scenarios and problems, allowing for a variety of approaches based on specific requirements and available data.

1.2.1 Supervised Learning

Supervised learning is the most popular type of machine learning, often used in applications like intrusion detection systems, where the system must be highly accurate in its predictions. In this category, a model learns from a labeled dataset in which the correct outputs are known for input vectors. While learning, the model adjusts its parameters to make its predictions similar to actual data. After training, it can predict outcomes for new data. There are various supervised learning algorithms

that can be employed based on the specifics of the data and the complexity of the problem:

1. **Logistic Regression:** Logistic regression is not utilised for regression issues; rather, it is used for classification. It is particularly useful for binary classification, such as determining if network activity is normal or an intrusion. The algorithm predicts the probability that a given input set belongs to a category.

2. **Decision Trees:** The decision process is modeled as a tree, where branches represent decision paths and leaf nodes represent outcomes. Decision trees are easy to interpret and are particularly useful for understanding the rules and conditions under which attacks occur. They can also be used in intrusion detection systems to minimize errors.

3. **K-Nearest Neighbors (KNN):** KNN is one of the simplest instance-based learning methods, where the output is class membership. An object is placed into the class most common among its neighbors by a majority of votes. This method is effective in intrusion detection systems where the similarity of behavior may indicate similar types of network traffic.

4. **Naive Bayes:** The Naïve Bayes classifier is predicated on Bayes' Theorem-based classification methods. It makes the assumption that a feature's existence in a class has nothing to do with the existence of any other feature. Now, that may sound really naive, but in reality, a Naïve Bayes classifier has really worked well for fairly complex real-world situations, such as spam detection and document classification. As such, this method can also be applicable in filtering out normal activities from potential threats within network traffic.

5. **Random Forest:** During the training phase, a large number of decision trees are constructed using this ensemble learning approach, which produces a class that is the mode of the classes of the individual trees. For classification, it has been found that ensembling increases classification accuracy by a big deal compared to a single decision tree. It is robust, easy to use, and therefore very suitable for intrusion detection systems.

6. **Support Vector Machines (SVM):** SVMs are flexible and powerful learning algorithms for classification and regression. They are based on finding a hyperplane that most effectively separates classes of objects in data into a margin as big as possible. In IDS, SVMs are used to effectively discriminate normal behavior from probable threats.

Supervised learning models need a large number of labeled data to effectively learn and train, which is why these are not applicable when labeling data is highly expensive and just not possible. However, the fact that they can provide very accurate predictions as long as they are trained with a sufficient and quality set of data makes them indispensable in security applications, especially in a structured environment like an IDS, in which previous knowledge about the kinds of attack can increment detection capabilities.

1.2.2 Unsupervised Learning

Unsupervised learning differs from supervised learning in that it learns from the unclassified and unlabeled dataset. That is, the system should learn to describe data, self-organize the properties, and understand the hidden structure to carry out proper inference. Unsupervised learning is a prime technique used to identify the hidden patterns or structures in data, which may not be directly visible. In particular, it finds a lot of use when data are abundant but unlabeled, such as in the scenario of baseline behaviors to be established in network systems in their early stages or in anomaly detection of intrusion detection systems.

1. **K-Means Clustering**: One easy and popular clustering approach is K-means clustering. The algorithm's goal is to cluster 'n' observations into 'k' clusters so that each observation is a prototype of the cluster and belongs to the cluster with the nearest mean. This makes the procedure very effective in segmenting data into clear groups that can point out the very unusual patterns or anomalies within the network traffic. For example, it could segment the network traffic according to similar behavior. Any significant deviations from such groups could be flagged for further investigation using K-means. An IDS, for example, could use this group information to classify network traffic with similar behavior. Any large deviations from such classes could be flagged for further investigation.

2. **Principal Component Analysis (PCA)**: PCA is a statistical dimensionality reduction technique that optimally preserves the maximum amount of variability. It allows for the identification of the directions of data variance maximization or principal components. This will be greatly useful in IDS, and the capability of PCA to linearly transform high-dimensional data into much fewer dimensions will be useful in exposing only the most significant features that account for changes in data. It helps reduce the number of dimensions, hence further helps to increase efficiency in other machine learning algorithms by reducing the computational overhead and helping data visualization easily to show unusual patterns.

Unsupervised learning can, therefore, help with intrusion detection systems using K-means clustering and PCA since these algorithms assist in the understanding and classification of very complicated, high-dimensional data in the absence of prior information related to the outputs. The algorithms automatically identify groups and patterns, specifically picking out all anomalies falling out of this particular paradigm to assist in identifying possible security breaches timely. In addition, unsupervised learning is adaptive to new data and is thus well suited to dynamic environments, such as network security, in which threat behaviors change continually.

1.2.3 Semi-Supervised Learning

Semi-supervised learning strikes a balance between supervised and unsupervised learning. This technique is applied in the scenario when most of the data is lying unlabeled, with just a small portion labeled, residing in the dataset. The task for semi-supervised learning, therefore, would be to tap into this huge volume of unlabeled data in order to get a better understanding of the underlying structure and distribution of the data, hence augment the performance of predictive models built with the limited labeled data.

The rationale behind semi-supervised learning is that the labeled and unlabeled data are drawn from the same distribution and, if used properly, the unlabeled data may provide an insight into the environment that is much better than that provided by the labeled data alone. This is particularly valid if acquiring labeled data is expensive and time-consuming, normally the case with intrusion detection in the IoT environment.

1.2.4 Reinforcement Learning

Reinforcement learning is another machine learning type that differs fundamentally from both supervised and unsupervised learning. A learning agent learns to make decisions by performing actions within an environment and receiving appropriate feedback through rewards or punishments. Such feedback helps the agent make decisions under which state conditions the best actions are, thus developing a policy of action upon direct interaction with the environment.

The hallmark of reinforcement learning is the balance that has to be struck between exploration (finding out new things) and exploitation (using known knowledge optimally). This forms the essence of drawing out optimal strategies that the agent could employ to maximize the sum of rewards over time. In general, RL is modeled as a Markov decision process, where the outcome is partly random and partly under the control of the decision-maker.

1.3 Deep Learning

Deep learning is a field in machine learning that uses deep neural networks, which are multi-level processing architectures for feature extraction at a significantly high level from the provided data. It has revolutionized computer vision, natural language processing, and audio recognition, among many others, and extended the resulting security applications to intrusion detection systems in IoT environments.

Deep learning models, specifically implemented on neural networks, are good at handling big and complicated datasets, extracting information, and learning features automatically without explicit manual extraction. The following are the most important types of neural networks used in deep learning:

1. **Artificial Neural Networks (ANNs):** ANNs are the foundation of deep learning. They consist of an input layer, a hidden layer, and an output layer. All are connected by adaptive weights. One of the most dominant characteristics possessed by ANNs is their pattern recognition. With this, it can be employed to try and specify or predict any uncommon network or user activity that might indicate a threat to security.

2. **Convolutional Neural Networks (CNNs):** Convolutional Neural Networks are designed to be applied specifically in data with a grid-like topology, for instance images. These networks can be applied to packet capture or any other grid-like data structure in security applications to reveal patterns indicative of malicious activity. Their ability to retain the spatial hierarchy in the data makes them very effective in contexts where the layout of the pattern could offer important clues about their nature.

3. **Recurrent Neural Networks (RNNs):** RNNs can handle sequential data, like time series or stream data. This proves invaluable to IDS, since the order in which network events happen can dramatically affect their effectiveness in detection and prediction. RNNs are uniquely capable of memorizing previous inputs, maintaining them in the 'memory' state, thereby helping make sense of the data flow over time.

4. **Long Short-Term Memory (LSTM) Networks:** An advanced kind of RNN called Long Short-Term Memory Networks (LSTMs) has the ability to learn lengthy data dependencies sequentially. The vanishing gradient problem, which occurs when an input's effect on a network's output diminishes exponentially or becomes unstable over time, used to plague traditional RNNs. LSTMs tackle this through the use of special gates that moderate the flow of information and are, therefore, very effective in determining complex multi-stage attacks in the intrusion detection systems, particularly where actions are spaced over extended periods.

5. **Bidirectional Encoder Representations from Transformers (BERT):** A cutting-edge approach of natural language processing called Bidirectional Encoder Representations from Transformers is constructed atop the Transformer architecture and use attention processes rather than recurrent sequence alignment. BERT can be fine-tuned for specific tasks such as the analysis of network protocols or understanding malicious scripts in an IDS environment. These are the reasons that underlie why it can understand intricate and complex patterns in data because it is able to look at the context of a token from both the left and the right side of sequences.

The deep learning concept and computational tool would then be applied in effecting the development of high-level and sophisticated intrusion detection systems. These models will be very good at detecting subtle and complex patterns of an attack, learning new threats, and adaptation to changing behavior without changing the reprogramming of the model. Hence, a high level of automation and adaptability is required in the fast-evolving field of network security and IoT.

1.3.1 Long Short-Term Memory (LSTM)

Long Short-Term Memory networks are specifically designed to overcome the weakness of the traditional RNNs, which is dealing with the vanishing and exploding gradient problems that are associated with sequences containing a large gap between important information.

The fundamental concept behind LSTMs is the cell state, which runs straight down the entire chain with minimal alteration, acting like a conveyor belt. Central to this mechanism are structures known as gates, which regulate the flow of information and maintain data on previous inputs.

- **Input gates:** Decide which values from the input should be used to update the memory.
- **Forget gates:** Allow the cell to forget outdated information no longer necessary for the LSTM to perform its task.
- **Output gates:** Determine what the next hidden state should be, which contains information on previous inputs.

Herein lies the power of LSTMs: that they can learn to keep or throw away information over long periods of time, later retrieving it when that helps to inform decisions in the present. This fact makes them exceptionally suited for application in network intrusion detection systems, as understanding the temporal context of the actions can become crucial for threat identification.

1.3.2 Bi-directional Long Short-Term Memory (Bi-LSTM)

Bi-LSTM is a simple extension of the classical LSTM by adding another layer to the feature information that the hidden layers convey in the opposite direction. This, in turn, thus moves bidirectionally so that the context is gathered from the past and future states. This further advances the better grasp of the model and better performance in tasks where the context in both directions is important.

Bi-LSTMs work well when, given the entire sequence, the network can make the most informed prediction at any given point. For example, in natural language

processing, this is useful in understanding the context that comes before and after a particular word or phrase.

In intrusion detection, using a Bi-LSTM will greatly reinforce the system's ability for pattern matching in order to discern complex cyber-attacks. For instance, it could understand the relation between the stages of attacks—reconnaissance, exploitation, data exfiltration—quite intricately and hence predict the subsequent steps or identify more accurately the type of attack being carried out in a given bidirectional LSTM.

In the proposed architecture, we have used LSTM and Bi-LSTM on three IDS benchmark datasets: NSL-KDD, UNSW-NB15, and CICIDS 2017.

1.4 Intrusion Detection Systems (IDS)

An Intrusion Detection System is a vital part of a cybersecurity infrastructure designed to detect unauthorized access, misuse, or a breach of a computer system. With the emerging and rapidly increasing complexity of cyber threats in environments like the Internet of Things and large-scale enterprises, the IDS should always detect and reduce any potential threats. This section presents the types, methodologies, and integration of machine learning techniques within IDS for a full understanding of the role they play in the protection of digital assets.

1.4.1 Types of Intrusion Detection Systems

IDS are broadly categorized into two types based on their monitoring approach:

Fig 1: Types of Intrusion Detection Systems

1. **Network Intrusion Detection Systems (NIDS):** This system operates by monitoring all network traffic for any known malicious or widely circulating threats. NIDS verifies the traffic stream between devices for anomalies and patterns deemed malignant. Normally, NIDS is deployed at some strategic point in the network to monitor inbound and outbound traffic across the network.

2. **Host-based Intrusion Detection Systems (HIDS):** These are installed on individual devices in a network. HIDS are implemented on individual devices within the network and observe both inbound and outbound communication from the device they are installed on. They also observe system interactions, such as file-system modifications and registry changes. Generally, a HIDS can give insight into the specific activities of a single host and provide detailed monitoring of operations performed by the system or its users.

1.4.2 Methodologies in Intrusion Detection

IDS methodologies can be broadly classified into three primary types based on how they detect intrusions:

1. **Signature-based Detection:** This is a method based on the use of pre-defined signatures of known threats, much like the working of antivirus software. It works by matching data patterns against the patterns in a database of known attack signatures and rules. While it is very effective against known threats, it cannot detect a new attack, known as a zero-day attack, which has not yet acquired a signature for detection.

2. **Anomaly-based Detection:** Anomaly-based detection, in contrast to signature-based systems, essentially tries to identify behavior that may be odd or deviate from the normal. In these systems, machine learning techniques try to build a model of normal behavior, and afterwards, any action not consistent with that model is flagged as a

potential threat. This allows the detection of novel or zero-day attacks that do not correspond to any existing signature.

3. **Stateful Protocol Analysis:** It is the process through which comprehension and adherence to a network protocol state is developed over the sequence of packets. The stateful protocol analysis should be used to detect anomalies in the intended use of the protocol, which may eventually become an indication of attack. This method can be made resource-intensive easily but guarantees the highest level of analysis of traffic flows and patterns.

1.4.3 Integration of Machine Learning in IDS

Machine learning and deep learning are applied in IDS, revolutionizing the traditional techniques of detection, in which the system learns from data and performs better over time. Huge datasets are analyzed by models of machine learning in discerning subtle patterns and anomalies that might represent a danger. The application of different models is as follows:

- **Supervised Learning for Signature Detection:** Machine learning algorithms, such as decision trees, SVMs, and neural networks, can be trained on labeled data with examples of malicious and benign activities, so the similar activities taking place in the operational data can be recognized.

- **Unsupervised Learning for Anomaly Detection:** lustering, PCA, and other techniques should help detect unusual patterns that do not correspond to usual traffic behavior and can actually help identify new types of cyber threats.

- **Reinforcement Learning for Adaptive Learning:** Reinforcement learning allows IDS to adapt to a changing network environment with time.

CHAPTER 2

LITERATURE REVIEW

The related work is this section is described below.

Patgiri R [1] Provide a study that focuses on developing an IDS, using the Random Forest and SVM algorithms on the NSL-KDD dataset.

The research that was carried out by Vinayakumar R [2] is based on deep learning models, and more specifically deep neural networks, employed to develop IDS with superior features in detecting cyber-attacks.

The study provided by Yulianto P [3] is based on the application of a machine learning method called AdaBoost to enhance the capabilities of the IDS system in augmenting the detection feature using the CICIDS 2017 dataset.

Chouhan N [4] have used a novel deep learning approach for detecting network anomaly by applying the deep CNN approach.

In the research presented by Kim [5], an advanced Intrusion Detection System (IDS) has been developed with deep learning techniques, namely a fusion of Convolutional Neural Networks (CNNs) with Long Short-Term Memory networks (LSTMs).

The study by Hussain J [6] is focusing on the implementation of a DNN model for intrusion detection in SDN.Using the strengths of CNN and LSTM for the detection of network intrusions.

Rajesh P [7] states that spatial-temporal features can be processed and used effectively.

Binbusayyis [8] did exploratory research in intrusion detection with an unsupervised deep learning approach, by fusing a convolutional autoencoder (CAE) with a one-class support vector machine (OCSVM).

A two-stage work has been presented by Mushtaq E [9], using auto-encoders for effective dimension reduction of features and LSTMs for accurate classification and prediction of network intrusion. In order to enhance the detection power of an intrusion detection system in network security

Halbouni [10] suggested a hybrid model that combines Long Short-Term Memory (LSTM) networks with Convolutional Neural Networks (CNNs).

Patil S [11] attempted to employ explainable artificial intelligence approaches to improve the interpretability and transparency of intrusion detection systems. In order to improve intrusion detection in Industrial Internet of Things (IIoT) networks.

A method combining Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) was described by Altunay H [12].

Using the Kyoto 2006+ dataset and information entropy measurement, the existing machine learning approaches for network intrusion detection were presented by Zaman M [13].

CNN was used by the IDS in Najar A [14] for detection and classification on DDoS attacks.

Table 2.1 discusses other works in the field of Information Security

Table 2.1: ML and DL techniques used in the Related field.

Reference	**Year**	**Purpose of the Paper**	**ML Techniques used**	**Result**	**Limitations**
[15]	2012	to create a more advanced antivirus engine that can extract system API calls, categorize and rank files according to security risk, and scan files to acquire knowledge and identify possible infections.	Random Forest	Achieved a classification accuracy of 99.5556%	While the method is very effective for enterprise networks, it is processor-heavy and may not be suitable for home users.
[16]	2019	To explore Machine Learning algorithms in detecting ransomware, focusing on feature selection and classification to improve ransomware prediction.	SVM, RF, DT, BN, ANN, LR	SVM achieved the highest 88.2% accuracy, RMSE of 0.179	The paper's limitations might include the selection of only five attributes from a large set, which could affect the model's generalizability

					and the challenges in handling high-dimensional data.
[17]	2019	To improve intrusion detection accuracy using anomaly analysis with the CICIDS-2017 dataset, comparing KNN and DNN methods.	KNN, DNN	KNN achieved an accuracy of 90.913%, recall of 91.283%, and precision of 90.302%, DNN achieved an accuracy of 96.427%, recall of 96.557%, and precision of 96.288%.	The paper compares KNN and DNN but does not explore other advanced Machine Learning or Deep Learning models, which might provide better or comparable results.
[18]	2020	Introducing a Fog computing-based intrusion detection model for Internet of Things (IoT) network security that uses a recurrent neural network that was trained using an improved backpropagation method to identify different kinds of intrusions.	RNN	Accuracy-92.18% Precision-90.23% FPR-9.8%	The computational demands of the model, particularly when deployed at scale in IoT environments, are not thoroughly addressed.
[19]	2020	To propose a distributed Deep Learning system for detecting web attacks,	CNNs, Word2Vec, FastText, M-	Accuracy-99.410 TPR-98.91%,	Challenges in handling unknown words

		specifically SQL injections, XSS, and command injections, by analyzing URLs on edge devices.	ResNet	DRN-99.55%	not seen during training, the need for frequent updates to adapt to new types of attacks and ensuring the system's effectiveness across different IoT environments.
[20]	2020	to suggest a hybrid Deep Learning model for effective intrusion detection in big data scenarios that combines WDLSTM and CNN.	CNN, WDLSTM	achieved an overall accuracy of 97.17% for binary classification and 98.43% for multi-classification on the testing data samples.	While the model performs well on specific datasets, its effectiveness across a broader range of attack types and scenarios remains to be fully demonstrated.
[21]	2020	Utilise Transfer Learning in Cybersecurity,Estimating Vulnerability for Exploitation Based on Description	BERT, LSTM	Ex-BERT achieved 91.12% accuracy and 91.82% precision in predicting vulnerability exploitability.	The paper suggests that future work could include developing an online Learning model to adapt to concept drift over time, indicating a limitation in the current approach's adaptability.

[22]	2020	To develop a novel intrusion detection and defense method for LR-DDoS attacks in edge computing environments using Deep convolutional neural networks (DCNN) and Q-Learning.	DCNN	higher detection accuracy and faster response time compared to other methods like SVMs, K-means, and SLNNs.	The paper acknowledges the need for improvement in detection performance when available data are sparse, indicating a potential limitation in the model's effectiveness with limited data.
[23]	2020	To improve classification performance in network intrusion detection systems, a multi-objective feature selection method utilizing NSGA-II and logistic regression is proposed.	(NSGA-II) for feature selection and Logistic Regression for classification, along with DT classifiers (C4.5 DT, RF and NB Tree) for testing the selected features.	The results showed better accuracy with binary-class datasets compared to multi-class datasets with 99.39% on CIC-IDS2017, 99.65 on NSL-KDD and 94.90% on UNSW-NB15 Dataset.	The paper involves challenges related to the dynamic nature of Android malware, potential overfitting due to the large feature set, and the framework's adaptability to new malware variants.
[24]	2020	To propose a perimeter intrusion detection system that utilizes MLP and quantum classifiers to enhance detection accuracy.	MLP and Quantum classifiers	The system achieved a high level of accuracy, with PIDS using MLP achieving	The paper involves the practical implementation of quantum classifiers and the scalability of

				98.96% accuracy.	the proposed system.
[25]	2021	To identify DDoS assaults that cause a Reduction of Quality (RoQ) by combining Machine Learning techniques with a unique method that combines Euclidean Distance MLP, and Fuzzy Logic.	MLP, KNN, SVM, MNB and an integrated approach using Fuzzy Logic, MLP, and Euclidean Distance.	MLP achieved the best classification results among the ML algorithms, with a F1-score up to 99.87% for real traffic.	The proposed approach using Fuzzy Logic, MLP, and Euclidean Distance exhibited a longer execution time, which might hinder real-time detection capabilities.
[26]	2021	to create MLDroid, a web-based framework that uses a variety of Machine Learning algorithms to analyze API calls and app permissions in order to identify malware on Android devices.	Random Forest	Achieved a malware detection rate of 98.8%.	The paper have limitations related to the balancing of the dataset, the generalisation of the model to other types of attacks or datasets, and computational requirements for real-time implementation.
[27]	2021	Using machine learning techniques to improve information security awareness among humans, focusing on classification and clustering based on questionnaire results.	Logistic Regression, KNN, SVM, NB, DT, RF, K-Means, agglomerative and DBSCAN	SVM model achieved the highest accuracy with a score of 99.7%.	Potential limitations might include the dataset's cultural and geographical specificity, the risk of overfitting due

					to the high dimensionality of questionnaire data, and the generalizability of the findings.
[28]	2022	To address IoT security by detecting webshells using ensemble Machine Learning approaches,.	K-Means, MLP, NB, DT, SVM, KNN and their ensemble counterparts for improved detection efficiency and accuracy.	Ensemble model achieved accuracy of 98.37%	The study doesn't Deeply delve into how these models adapt over time.
[29]	2022	The study aims to leverage ML algorithms for automated DDoS attack detection.	RF, KNN, SVM, ANN	The paper reports high classification accuracy for every tested ML algorithm, often achieving F1-scores above 98%	While the paper discusses real-time detection, the scalability and efficiency of these solutions in diverse and larger network environments require further exploration.
[30]	2023	To build a strong intrusion detection system using ensemble-based Machine Learning techniques in order to improve network security.	Ensemble Methods	The model achieved more than 99% accuracy across various datasets with ensemble methods,	The potential limitations could involve the adaptability of the model to unseen types of attacks

CHAPTER 3
METHODOLOGY

In this section of methodology of machine and deep learning for intrusion detection in IoT devices we will majorly focus on implementation part. Initially we discuss about the available datasets and compare them on the basis of features, records, data source and description. Then we will discuss about the architecture of proposed LSTM and Bi-LSTM model along with major hyper parameters.

Table 3.1 : Various Standard Datasets for Intrusion Detection

Dataset	Year	Total Records	Feature	Attack Types	Source	Description
KDD Cup 99	1999	4.9 million	41	4 types	DARPA	Among the most established and used datasets for evaluating intrusion detection systems.
Kyoto 2006	2006	Varies daily	14	Multiple	Kyoto Univ.	Tracks and collects real traffic data along with honeypots and darknets since 2006.
NSL-KDD	2009	125,973	41	4 types	DARPA	Improved version of the KDD'99 dataset with duplicate records removed to prevent biased learning.
UNSW-NB15	2015	2.5 million	49	10 types	UNSW	Features modern attack types, generated from a mix of real normal activities and synthetic attacks.
CICIDS 2017	2017	2.8 million	80	14 types	CIC	A modern dataset with a variety of attack scenarios, including the latest attacks.
CICIDS 2018	2018	16 million	80	15 types	CIC	Continues from CICIDS 2017 with more data and attack types, reflecting more recent network environments.

3.1 Intrusion Detection Datasets used

For effective training and evaluation of machine learning models in the realm of network security, specifically intrusion detection, high-quality datasets are essential. Three prominent datasets used in this field are NSL-KDD, UNSW-NB15, and CICIDS 2017. Each dataset has distinct features and has been employed to benchmark the performance of advanced machine learning models like LSTM and Bi-LSTM for binary classification tasks. Below is a detailed description of each dataset along with the specific application of LSTM and Bi-LSTM models.

3.1.1 NSL-KDD

The KDD'99 dataset, which was first developed for the Third International Knowledge Discovery and Data Mining Tools Competition, has been enhanced to become the NSL-KDD dataset[31]. It tackles some of the intrinsic issues with the KDD'99 dataset, such duplicate entries that might skew a machine learning model's learning process. The NSL-KDD dataset includes a set of data points that are reasonably selected to avoid the aforementioned issues, allowing researchers to produce more effective intrusion detection models.

Features and Structure:

Number of Features: Each record in the dataset has 41 features, including content features inside a connection that are recommended by domain expertise, traffic features computed using a two-second time window, and fundamental aspects of individual TCP connections.

Labels: This dataset supports binary classification since the instances are labeled as normal or anomalous.

Model Application:

Binary classification, through the classification of network activities into normal activities or potential threats, is performed using both LSTM and Bi-LSTM models applied over the NSL-KDD dataset. The models relied on the intrinsic sequential features in network traffic data to capture the time dependency and patterns that could possibly imply intrusive behavior.

3.1.2 UNSW-NB15

The UNSW-NB15 dataset is developed by the Cyber Range Lab of the Australian Centre for Cyber Security with the aim of providing a comprehensive dataset that incorporates modern attack activities[32]. This differentiates the dataset from older datasets, which usually lack this modern attack activity. It is used in training and testing intrusion detection systems today.

Features and Structure:

Number of Features: It includes a total of 49 features, which contain flow-based features and additional features from contemporary protocols like HTTPS and SSH.

Labels: milarly to the NSL-KDD, it has data points labeled and identified as either 'normal' or 'attack', which gives it a target for a binary classification task.

Model Application:

LSTM and Bi-LSTM applied to the UNSW-NB15 dataset provide dynamism in learning and adapt to newer and even more complex patterns of attack that might not have been represented by old datasets. These models harness the power of the diverse feature sets contained in the dataset to distinguish normal network traffic from the malicious traffic.

3.1.3 CICIDS 2017

The CICIDS 2017 dataset was created by the Canadian Institute for Cybersecurity and is among the most realistic datasets that can be used for intrusion detection systems, revealing in a real-world situation common attack scenarios like DDoS, DoS, Heartbleed, and many others[33]. It is this diversity in representation of the traffic and attacks that has been specially appreciated in the dataset.

Features and Structure:

Number of Features: It contains the entire features of traffic that would be existing in a real network, such as flow duration, packet length statistics, and flag types.

Labels: The labeling classifies every record as 'normal' or as belonging to a certain attack category, which can be used for developing detailed binary or multi-class classification models.

Model Application:

Detection of very intricate patterns of attacks and anomalies is provided by the application of LSTM and Bi-LSTM models over the CICIDS 2017 dataset. It is attributed to the learning and adaptation capability of the sequential temporal properties caught by these models in the process of detection of the sophisticated and diversified nature of the attacks represented in the dataset.

Summarizing, all these datasets—NSL-KDD, UNSW-NB15, and CICIDS 2017—represent different environments for the training of ML models, including but not limited to LSTM and BiLSTM. The application of such models in the binary classification task has greatly advanced the area of intrusion detection with the help of tools that can learn and adapt from the network environment and attacks.

3.2 Data Preprocessing

Data preprocessing is, by far, one of the key stages in any machine learning workflow. Data quality and the format in which intrusion detection data is available can considerably affect the performance of the models developed. Below are key preprocessing steps we have applied for the following datasets: NSL-KDD, UNSW-NB15, and CICIDS 2017.

3.2.1 Handling Missing Values

In both UNSW-NB15 and CICIDS 2017 datasets, missing values may have arisen from any fault in data collection or data processing; therefore, such gaps must be treated appropriately. As is commonly the case, missing numerical values are imputed by the median or mean of their respective columns. This is taken as a first choice since it maintains the central tendency without being skewed by some extremely large or low values, as could be the case with the mean in the presence of such values.

3.2.2 Data Integration

Such datasets, for instance UNSW-NB15, are separated by parts and need to be merged into one single dataset. This is done by appending numerous DataFrame objects into one so that none of the data points go missing in the analysis and model training process. This step is, again, very important in keeping data consistency and completeness.

3.2.3 Label Encoding

Binary classification tasks function in two clear-cut categories, usually 'normal' and 'attack'. Label encoding converts them into binary format, mostly '0' and '1', that once again is a format that is ready to be processed by the model. This shall be seen in the datasets considered, as the labels are transformed in such a way that it could be easily read or interpreted by the learning algorithms.

3.2.4 Feature Scaling

This is done so that the machine learning model treats all the features equally. It becomes especially crucial when a number of variables that are of different scales are combined into an effective learning data source of the model. Techniques like MinMaxScaler and StandardScaler, as indicated in the datasets, help in normalizing or standardizing the features so that each feature contributes evenly in analysis.

3.2.5 Handling Categorical Variables

Most datasets contain categorical features that need to be converted into a numerical format for processing by machine learning algorithms. One-hot encoding is a method that converts a categorical variable into a series of binary variables. This transformation is important for the model to be able to understand and evaluate the input data without any order relationship, as such a relationship may mislead the learning algorithm.

3.2.6 Data Reshaping

Data passed into this model, if using a deep learning or LSTM/Bi-LSTM model, will need reshaping to represent the structure of the model. In general, reshaping is performed to represent the dataset with dimensions on batches, time steps, and features. Data reshaping is what one does to prepare the dataset in the best way to operate easily on sequences and time-series data inside the question model, thus learning underlying temporal patterns and dependencies bound to be found in the dataset.

All these preprocessing steps are important to prepare the data for effective and efficient analysis using machine learning and deep learning models in the domain of intrusion detection. The importance of each of the steps lies in the fact that data shall meet the specifications of particular algorithms in use, making them generally accurate and effective in their performance for the proposed systems of intrusion detection.

3.3 Proposed Models

The proposed models will utilize deep learning for enhancing intrusion detection systems adopted by the architectures of the Long Short-Term Memory and Bi-directional Long Short-Term Memory networks. These models will be capable of capturing important temporal and sequential dependencies in network traffic data, which are important for the detection of patterns indicative of cyber threats. This section provides a detailed theoretical background and architecture regarding the utilization of the NSL-KDD, UNSW-NB15, and CICIDS 2017 datasets.

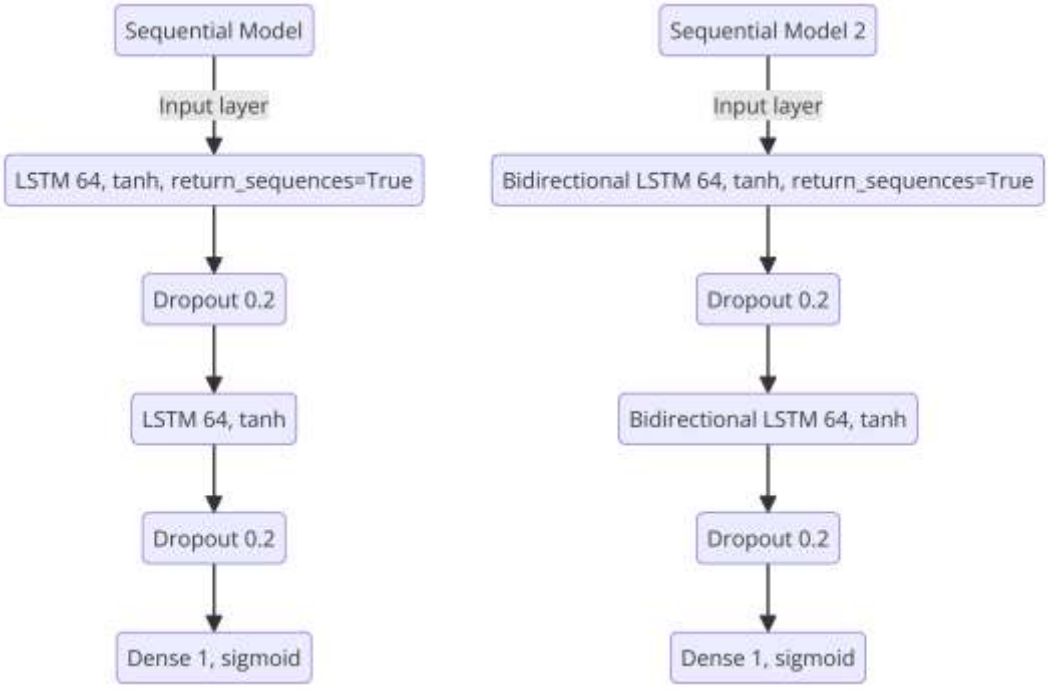

Fig 3.1 : Architecture of proposed Model

3.3.1 LSTM Model

Architecture:

The LSTM model follows a sequential approach, which indicates its ability to consider data inputs one at a time[34]. The architecture includes:

Input Layer: Input Layer: The input shape is adapted to reflect re-shape of the feature set into the format (1, number of features) required for sequence processing..

LSTM Layers: The proposed architecture has two LSTM layers. The first layer of LSTM is defined with the argument return_sequences=True so that this layer can pass its output as a sequence to the next layer, instead of flattening the output, thereby preserving temporal information across the network.

The first LSTM layer is of size 64 with the activation function 'tanh'. An activation function that has an upper hand in not resulting in being binary is, thereby, good for modeling nonlinearities, which are key to learning complex data patterns.

This is followed by a second layer of LSTM, again with 64 units, which, however, does not return sequences, thus serving to summarize the features learned from the input sequence.

Dropout Layers: Following every LSTM layer, a Dropout layer of rate 0.2 is placed. This makes the input units zero at random during each update in the training phase and therefore prevents overfitting and increases the generalization of the model.

Output Layer: It consists of a one-unit Dense layer where the activation function is 'sigmoid'; it is built for binary classification that classifies whether the input data point is 'normal' or 'attack'.

Functionality:

LSTM is more fitted to sequences in which the order and context of events play a role. With their internal states and gates, LSTMs can either remember or forget

information, making them quite applicable in IDS, as an attack will typically be contextually linked to previous events in the data stream.

3.3.2 Bi-LSTM Model

Architecture:
The Bi-LSTM increases the possibilities of the standard LSTM by providing bidirectional processing and thus allows for the following:

Input Layer: This layer is somewhat akin to the LSTM in that it reshapes the input into a format suitable for sequence processing.

Bi-directional LSTM Layers:
The first Bi-LSTM layer exactly mirrors the LSTM layer of the setup but processes data in both directions[35]. The dual pathway ensures that the model learns information about the past and future contexts at the same time, which, in the case of uni-directional processing, significantly helps in capturing the information that could be lost.

The second Bi-LSTM layer processes the bidirectional sequence data without returning, so learned features are consolidated across the temporal dimension.

Dropout Layers: These layers also prevent overfitting, similar to the LSTM model. They randomly set input units at every update during training to 0 and therefore add to model generalization.

Output Layer: A sigmoid activation function is used for binary classification, a function that makes a decision of whether the data point should be classified as "normal" or "attack".

Functionality:

Bi-LSTMs become very effective, especially when the contexts of both precedents and antecedents, relative to a data point, are required in order to carry out accurate predictions. Such use cases create value within an intrusion detection system by allowing the system to use the information of both the preceding and following packets or connections to classify more effectively and recognize clearer patterns.

Application in IDS

The LSTM and Bi-LSTM models are used on the NSL-KDD, UNSW-NB15, and CICIDS 2017 datasets for binary classification tasks between normal and malicious activities. They sequence and thus form good detectors of cyber threats capable of detecting complex patterns or anomalies in cyber threats. In addition, these models are highly applicable in the processing and learning from time-series data and can be used for dynamic and complex systems with intrusive activities in the network systems. The application targets accurate and efficient intrusion detection systems applying deep learning to make them adaptable to the changing nature of cyber threats.

Table 3.2: Purposed Model layers and Parameters for LSTM layer

Layer Name	**Parameter Name**	**Parameter Value**
LSTM	units	32/64
LSTM	activation	'tanh'
LSTM	return_sequences	True
LSTM	input_shape	(1, X_train.shape[2])
Dropout	rate	0.2
LSTM	units	32/64
LSTM	activation	'tanh'
LSTM	return_sequences	False
Dropout	rate	0.2
Dense	units	1
Dense	activation	'sigmoid'

Table 3.3: Purposed Model layers and Parameters for Bi-LSTM layers

Layer Name	**Parameter Name**	**Parameter Value**
Bi-LSTM	units	32/64
Bi-LSTM	activation	'tanh'
Bi-LSTM	return_sequences	True
Bi-LSTM	input_shape	(1, X_train.shape[2])
Dropout	rate	0.2
Bi-LSTM	units	32/64
Bi-LSTM	activation	'tanh'
Bi-LSTM	return_sequences	False
Dropout	rate	0.2
Dense	units	1
Dense	activation	'sigmoid'

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Experimental Setup

The purposed model is implemented on KAggle on Google Compute Engine backend (GPU). The system was having a intel core i7(6^{th} gen) processor with disk space of around 100 GB.Hardware acceleration of GPU was provided for faster execution and training of model.

4.2 Evaluation Metrics

Evaluation metrics are used in deep learning to assess a model's performance. We make use of classification measures such as accuracy, precision, recall, and F1 score to assess the performance of our models. An analysis of the performance may be done with a confusion matrix. It is a binary classification matrix, made out of 2x2 tables.

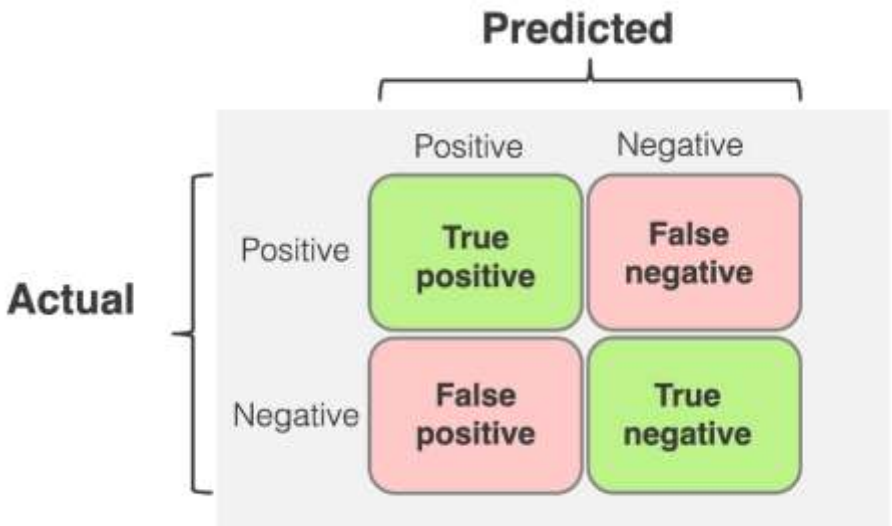

Fig 4.1: Diagram of Confusion Matrix

- **True Positive (TP):** The class that our model predicted to be "malicious" actually is malicious.

- **True Negative (TN):** The real class is "non-malicious," but our model projected it to be "non-malicious."

- **False Positive (FP):** The real class is "non-malicious," although our model projected it to be "malicious."

- **False Negative (FN):** The real class is "malicious," but our model projected it to be "non-malicious."

4.3 Result Analysis

The performance of the LSTM and Bi-LSTM models on the NSL-KDD, UNSW-NB15, and CICIDS 2017 datasets provides a comprehensive view of the effectiveness of these deep learning techniques in detecting network intrusions. Here, we analyze the results obtained from these models to understand their capabilities and areas of application.

4.3.1 NSL-KDD

LSTM Model Results:

Precision of 98.36% for normal class and 97.24% for attack class, Recall of 96.57% for Normal class and 98.50% for attack class resulting an F1 Score of 97.66% for Normal class and 97.87% for Attack class and accuracy of 97.77%

Bi-LSTM Model Results:

Precision of 98.90% for normal class and 97.85% for attack class, Recall of 97.64% for Normal class and 99.00% for attack class resulting an F1 Score of 98.27% for Normal class and 98.42% for Attack class and accuracy of 98.35.

Table 4.1 Results Calculated for NSL-KDD Dataset

	LSTM		Bi-LSTM	
	Normal Class	Attack Class	Normal Class	Attack Class
Accuracy	97.77%		98.35%	
Precision	98.36%	97.24%	98.90%	97.85%
Recall	96.57%	98.50%	97.64%	99.00%
F1-Score	97.87%	97.77%	98.27%	98.42%

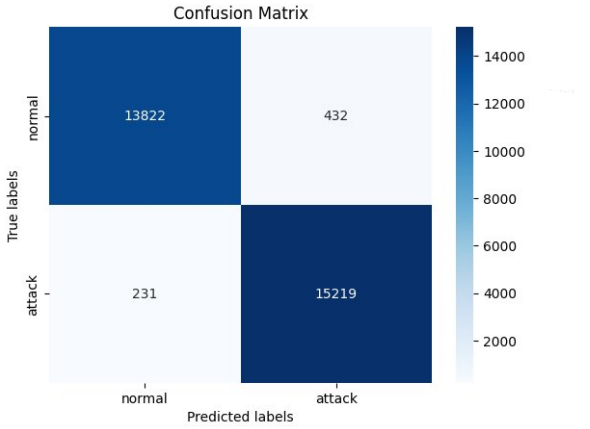

Fig 4.2: Confusion Matrix of NSL-KDD on LSTM

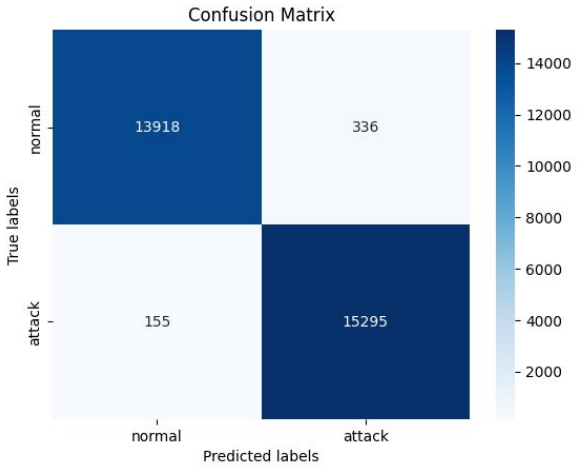

Fig 4.3: Confusion Matrix of NSL-KDD on Bi-LSTM

The Bi-LSTM model's superior performance in NSL-KDD is likely due to its ability to capture additional contextual information from the sequence data, which is pivotal in intrusion detection scenarios.

4.3.2 UNSW-NB15

LSTM Model Results:

Precision of 98.62% for normal class and 99.99% for attack class, Recall of 99.99% for Normal class and 98.61% for attack class resulting an F1 Score of 99.30% for Normal class and 99.29% for Attack class and accuracy of 99.30%

Bi-LSTM Model Results:

Precision of 98.56% for normal class and 100.00% for attack class, Recall of 100.00% for Normal class and 98.55% for attack class resulting an F1 Score of 99.27% for Normal class and 99.27% for Attack class and accuracy of 99.27%.

Table 4.2 Results Calculated for UNSW-NB15 Dataset

	LSTM		Bi-LSTM	
	Normal Class	Attack Class	Normal Class	Attack Class
Accuracy	99.30%		99.27%	
Precision	98.62%	99.99%	98.56%	100.00%
Recall	99.99%	98.61%	100.00%	98.55%
F1-Score	99.30%	99.29%	99.27%	99.27%

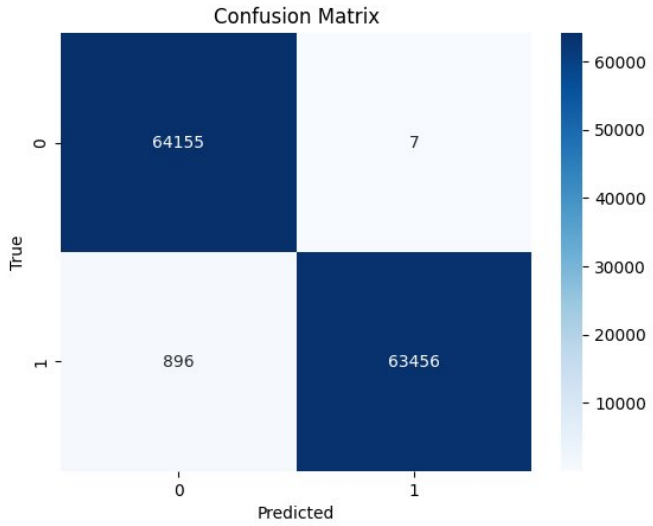

Fig 4.4: Confusion Matrix of UNSW-NB15 on LSTM

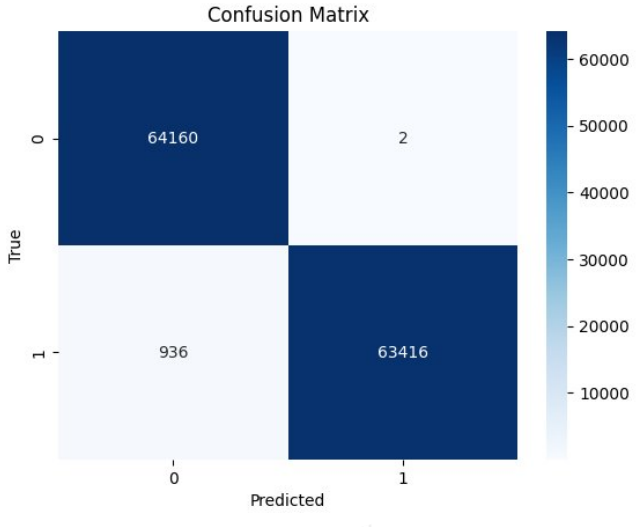

Fig 4.5 Confusion Matrix of UNSW-NB15 on Bi-LSTM

Both LSTM and Bi-LSTM models show remarkable effectiveness on the UNSW-NB15 dataset. The small difference in performance metrics between the two models suggests that both are well-suited for datasets with modern and diverse attack simulations.

4.3.3 CICIDS-2017

LSTM Model Results:

Precision of 97.73 % for normal class and 99.14% for attack class, Recall of 99.15 for Normal class and 97.70% for attack class resulting an F1 Score of 98.44% for Normal class and 98.42% for Attack class and accuracy of 98.43%

Bi-LSTM Model Results:

The analysis shows significant performance metrics across both LSTM and Bi-LSTM models. The precision for the normal class stands at 97.64%, and for the attack class, it is 99.44%. The recall rates are 99.45% for the normal class and 97.60% for the attack class, leading to F1 scores of 98.54% for the normal class and 98.51% for the attack class, with an overall accuracy of 98.53%.

Table 4.3 Results Calculated for CICIDS-2017 Dataset

	LSTM		Bi-LSTM	
	Normal Class	Attack Class	Normal Class	Attack Class
Accuracy	99.43%		99.53%	
Precision	97.73%	99.14%	97.64%	99.44%
Recall	99.15%	97.70%	99.45%	97.60%
F1-Score	98.44%	98.42%	98.54%	98.51%

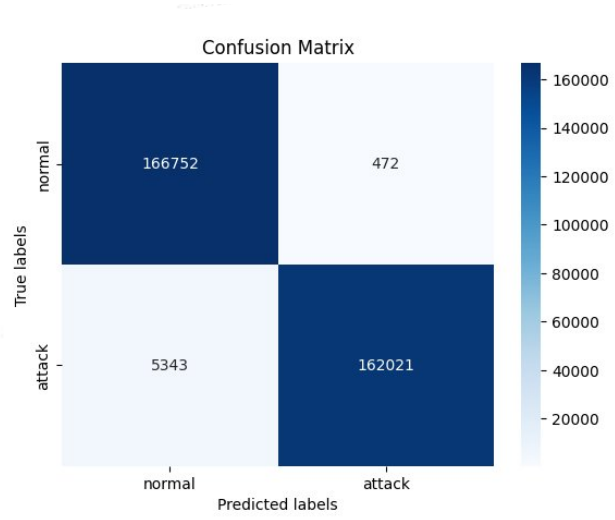

Fig 4.6: Confusion Matrix of CICIDS 2017 on LSTM

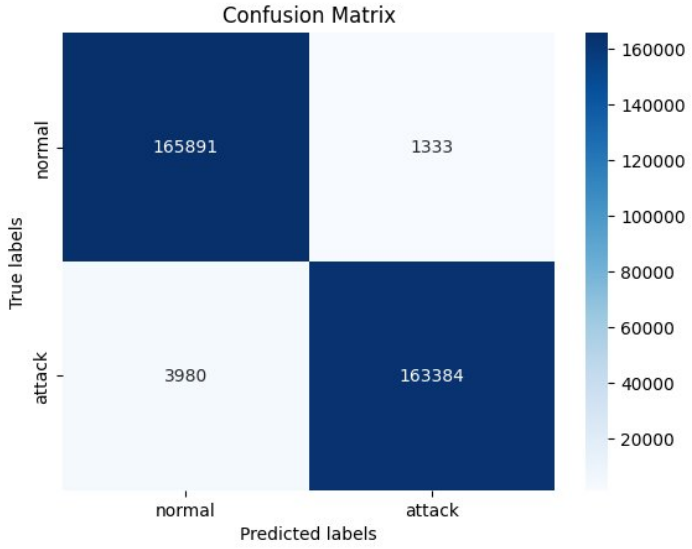

Fig 4.7: Confusion Matrix of CICIDS 2017 on Bi-LSTM

The results indicate that while both models achieved truly high performance for the CICIDS-2017 dataset, the improvements gained by the Bi-LSTM model outshone them in terms of overall results. This performance edge is likely to have resulted from the bidirectionality of the Bi-LSTM to use context both from past and future data points in this dataset's complex and varied attack scenarios.

4.3.4 Summary

In general, the performance difference between the Bi-LSTM and LSTM models on all test datasets—NSL-KDD, UNSW-NB15, and CICIDS 2017—remained quite marginal, except in the measures of recall and global F1 score. In view of this, adding extra context brought by bidirectionality of the data processing seems helpful in increasing the intrusion detection capability of such models, especially in cases of complex intrusion scenarios. The results, thus, put in relief the potential utilization of advanced deep learning models such as LSTM or BiLSTM for the improvement of cybersecurity measures in heterogeneous network environments.

CHAPTER 5
CONCLUSION AND FUTURE SCOPE

LSTM and Bi-LSTM-based approaches are very effective techniques for the identification of network intrusion instances over a set of benchmark datasets. These models capture the sequential and temporal dependencies of network data for the discovery of subtle and complex patterns of attacks, hence leading to increased detection accuracy of threats. Results show that the Bi-LSTM models outperform the standard LSTM models with just minor, though positive, effect on the overall improved detection when dealing with bidirectional data sequences. This work has shown the potential of deep learning techniques in making the current IDS systems more robust and reliable.Looking ahead, there are a number of refinements and extensions that could push performance and applicability even further for machine learning models in cybersecurity. This will involve more cohesive integration with other technological advances, tightening designs on model architecture, and escalating training on new typologies of threats in order for security systems to operate effectively against emerging cyber threats.Models explored in this work could be further trained on bigger and more diverse datasets, such as CICIDS 2018 and beyond**.**Binary classification proves useful in distinguishing normal activities from the malicious ones. Multi-class classification will take place in categorizing network attacks of different types and, therefore, allow nuancing the responses to the different levels and types of threats, improving the security posture in network systems.Data augmentation may be one of the useful strategies for growing the volume and quality of training datasets, especially around underrepresented types of attacks. Techniques such as synthetic minority over-sampling or the generation of artificial data samples come into play as a means of balancing the dataset, which can be useful for making training across the different categories of attacks more uniform.Beyond this, further work could be done to develop real-time intrusion detection and response systems, whereby threats would have immediate response systems in place to counter them.

REFERENCES

[1] R. Patgiri, U. Varshney, T. Akutota, and R. Kunde, “An Investigation on Intrusion Detection System Using Machine Learning,” in 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India: IEEE, Nov. 2018, pp. 1684–1691. doi: 10.1109/SSCI.2018.8628676.

[2] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, “Deep Learning Approach for Intelligent Intrusion Detection System,” IEEE Access, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[3] Yulianto, P. Sukarno, and N. A. Suwastika, “Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset,” J. Phys.: Conf. Ser., vol. 1192, p. 012018, Mar. 2019, doi: 10.1088/1742-6596/1192/1/012018.

[4] N. Chouhan, A. Khan, and H.-R. Khan, “Network anomaly detection using channel boosted and residual learning based deep convolutional neural network,” Applied Soft Computing, vol. 83, p. 105612, Oct. 2019, doi: 10.1016/j.asoc.2019.105612.

[5] Kim, M. Park, and D. H. Lee, “AI-IDS: Application of Deep Learning to Real-Time Web Intrusion Detection,” IEEE Access, vol. 8, pp. 70245–70261, 2020, doi: 10.1109/ACCESS.2020.2986882.

[6] J. Hussain and V. Hnamte, “A Novel Deep Learning Based Intrusion Detection System : Software Defined Network,” in 2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Zallaq, Bahrain: IEEE, Sep. 2021, pp. 506–511. doi: 10.1109/3ICT53449.2021.9581404.

[7] P. Rajesh Kanna and P. Santhi, “Unified Deep Learning approach for Efficient Intrusion Detection System using Integrated Spatial–Temporal Features,” Knowledge-Based Systems, vol. 226, p. 107132, Aug. 2021, doi: 10.1016/j.knosys.2021.107132.

[8] Binbusayyis and T. Vaiyapuri, “Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM,” Appl Intell, vol. 51, no. 10, pp. 7094–7108, Oct. 2021, doi: 10.1007/s10489-021-02205-9.

[9] E. Mushtaq, A. Zameer, M. Umer, and A. A. Abbasi, “A two-stage intrusion detection system with auto-encoder and LSTMs,” Applied Soft Computing, vol. 121, p. 108768, May 2022, doi: 10.1016/j.asoc.2022.108768.

[10] Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, “CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System,” IEEE Access, vol. 10, pp. 99837–99849, 2022, doi: 10.1109/ACCESS.2022.3206425.

[11] S. Patil et al., “Explainable Artificial Intelligence for Intrusion Detection System,” Electronics, vol. 11, no. 19, p. 3079, Sep. 2022, doi: 10.3390/electronics11193079.

[12] H. C. Altunay and Z. Albayrak, “A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks,” Engineering Science and Technology, an International Journal, vol. 38, p. 101322, Feb. 2023, doi: 10.1016/j.jestch.2022.101322.

[13] M. Zaman and C.-H. Lung, “Evaluation of machine learning techniques for network intrusion detection,” in NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, Taiwan: IEEE, Apr. 2018, pp. 1–5. doi: 10.1109/NOMS.2018.8406212.

[14] A. Najar and M. N. S., “A Robust DDoS Intrusion Detection System Using Convolutional Neural Network,” Computers and Electrical Engineering, vol. 117, p. 109277, Jul. 2024, doi: 10.1016/j.compeleceng.2024.109277.

[15] P. Singhal, “Malware Detection Module using Machine Learning Algorithms to Assist in Centralized Security in Enterprise Networks,” IJNSA, vol. 4, no. 1, pp. 61–67, Jan. 2012, doi: 10.5121/ijnsa.2012.4106.

[16] U. Adamu and I. Awan, “Ransomware Prediction Using Supervised Learning Algorithms,” in 2019 7th International Conference on Future Internet of Things and Cloud (FiCloud), Istanbul, Turkey: IEEE, Aug. 2019, pp. 57–63. doi: 10.1109/FiCloud.2019.00016.

[17] K. Atefi, H. Hashim, and M. Kassim, “Anomaly Analysis for the Classification Purpose of Intrusion Detection System with K-Nearest Neighbors and Deep Neural Network,” in 2019 IEEE 7th Conference on Systems, Process and Control (ICSPC), Melaka, Malaysia: IEEE, Dec. 2019, pp. 269–274. doi: 10.1109/ICSPC47137.2019.9068081.

[18] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, “Deep recurrent neural network for IoT intrusion detection system,” Simulation Modelling Practice and Theory, vol. 101, p. 102031, May 2020, doi: 10.1016/j.simpat.2019.102031.

[19] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, “A Distributed Deep Learning System for Web Attack Detection on Edge Devices,” IEEE Trans. Ind. Inf., vol. 16, no. 3, pp. 1963–1971, Mar. 2020, doi: 10.1109/TII.2019.2938778.

[20] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, “A hybrid deep learning model for efficient intrusion detection in big data environment,” Information Sciences, vol. 513, pp. 386–396, Mar. 2020, doi: 10.1016/j.ins.2019.10.069.

[21] J. Yin, M. Tang, J. Cao, and H. Wang, “Apply transfer learning to cybersecurity: Predicting exploitability of vulnerabilities by description,” Knowledge-Based Systems, vol. 210, p. 106529, Dec. 2020, doi: 10.1016/j.knosys.2020.106529.

[22] Z. Liu, X. Yin, and Y. Hu, “CPSS LR-DDoS Detection and Defense in Edge Computing Utilizing DCNN Q-Learning,” IEEE Access, vol. 8, pp. 42120–42130, 2020, doi: 10.1109/ACCESS.2020.2976706.

[23] C. Khammassi and S. Krichen, “A NSGA2-LR wrapper approach for feature selection in network intrusion detection,” Computer Networks, vol. 172, p. 107183, May 2020, doi: 10.1016/j.comnet.2020.107183.

[24] A. Thirumalairaj and M. Jeyakarthic, “Perimeter Intrusion Detection with Multi Layer Perception using Quantum Classifier,” in 2020 Fourth International Conference on Inventive Systems and Control (ICISC), Coimbatore, India: IEEE, Jan. 2020, pp. 348–352. doi: 10.1109/ICISC47916.2020.9171159.

[25] V. D. M. Rios, P. R. M. Inácio, D. Magoni, and M. M. Freire, “Detection of reduction-of-quality DDoS attacks using Fuzzy Logic and machine learning algorithms,” Computer Networks, vol. 186, p. 107792, Feb. 2021, doi: 10.1016/j.comnet.2020.107792.

[26] A. Mahindru and A. L. Sangal, “MLDroid—framework for Android malware detection using machine learning techniques,” Neural Comput & Applic, vol. 33, no. 10, pp. 5183–5240, May 2021, doi: 10.1007/s00521-020-05309-4.

[27] S. Siwi and S. Fitri, “Implementation of machine learning for human aspect in information security awareness,” J Appl Eng Science, vol. 19, no. 4, pp. 1126–1142, 2021, doi: 10.5937/jaes0-28530.

[28] B. Yong et al., “Ensemble machine learning approaches for webshell detection in Internet of things environments,” Trans Emerging Tel Tech, vol. 33, no. 6, p. e4085, Jun. 2022, doi: 10.1002/ett.4085.

[29] F. Musumeci, A. C. Fidanci, F. Paolucci, F. Cugini, and M. Tornatore, “Machine-Learning-Enabled DDoS Attacks Detection in P4 Programmable Networks,” J Netw Syst Manage, vol. 30, no. 1, p. 21, Jan. 2022, doi: 10.1007/s10922-021-09633-5.

[30] [1] Md. A. Hossain and Md. S. Islam, “Ensuring network security with a robust intrusion detection system using ensemble-based machine learning,” Array, vol. 19, p. 100306, Sep. 2023, doi: 10.1016/j.array.2023.100306.

[31] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in 2009 IEEE Symposium on Computational Intelligence for Security

and Defense Applications, Ottawa, ON, Canada: IEEE, Jul. 2009, pp. 1–6. doi: 10.1109/CISDA.2009.5356528.

[32] 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), Canberra, Australia: IEEE, Nov. 2015, pp. 1–6. doi: 10.1109/MilCIS.2015.7348942.

[33] R. Panigrahi and S. Borah, “A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems,” International Journal of Engineering.

[34] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[35] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” IEEE Trans. Signal Process., vol. 45, no. 11, pp. 2673–2681, Nov. 1997, doi: 10.1109/78.650093.

List of Publications

[1] Sanchit Agarwal and Pawan singh mehra, “A survey of various Machine Learning and Deep Learning Approaches in the field of Information Security”, communicated and accepted at 1st International Conference on Advances in Computing, Communication and Networking- ICAC2N, 16th - 17th December 2024, Greater Noida, India

[2] Sanchit Agarwal and Pawan singh mehra, “Evaluating LSTM and Bi-LSTM for Binary Classification in Intrusion Detection System in IoT”, communicated and accepted at 1st International Conference on Applied Artificial Intelligence and Machine learning- ICAAIML, 30th – 31st August 2024, Hyderabad, Telangana, India

Gmail Sanchit Agarwal <sanchitagarwal969@gmail.com>

Acceptance Notification 1st IEEE ICAC2N-2024 & Registration: Paper ID 658 @ ITS Engineering College, Greater Noida

1 message

Microsoft CMT <email@msr-cmt.org> Thu, May 9, 2024 at 8:12 AM
Reply-To: "Dr. Vishnu Sharma" <vishnu.sharma@its.edu.in>
To: Sanchit Agarwal <sanchitagarwal969@gmail.com>

Dear Sanchit Agarwal,
Delhi Technological University,New Delhi

Greetings from ICAC2N-2024 ...!!!

Congratulations....!!!!!

On behalf of the ICAC2N-2024 organising Committee, we are delighted to inform you that the submission of “Paper ID- 658 “ titled “ A survey of various Machine Learning and Deep Learning Approaches in the field of Information Security “ has been accepted for presentation and further publication with IEEE at the ICAC2N- 24. All accepted papers will be submitted for inclusion into IEEE Xplore subject to meeting IEEE Xplore’s scope and quality requirements.

Registration/Fee Payment related details are available at https://icac2n.in/register.

For early registration benefit please pay your fee and complete your registration by clicking on the following Link: https://forms.gle/E7RuvuQQPxPZQnJU6 by 15 May 2024.

You are directed to ensure incorporating following points in your paper while completing your registration:

Comments:
The topic chosen "A survey of various Machine Learning and Deep Learning Approaches in the field of Information Security" is interesting and relevant.
Formatting of paper is not proper. Paper must be strictly in IEEE template.
Abstract must be clear and precise.
Add a comparison table with the similar work carried out in this field with latest references.
Conclusion and result section must be more descriptive.
All references must be properly cited in content and should be in proper format.

Note:
1. All figures and equations in the paper must be clear.
2. Final camera ready copy must be strictly in IEEE format available on conference website.
3. Transfer of E-copyright to IEEE and Presenting paper in conference is compulsory for publication of paper in IEEE.
4. If plagiarism is found at any stage in your accepted paper, the registration will be cancelled and paper will be rejected and the authors will be responsible for any consequences. Plagiarism must be less then 15% (checked through Turnitin).
5. Change in paper title, name of authors or affiliation of authors will not be allowed after registration of papers.
6. Violation of any of the above point may lead to rejection of your paper at any stage of publication.
7. Registration fee once paid will be non refundable.

If you have any query regarding registration process or face any problem in making online payment, write us at icac2n.ieee@gmail.com.

Regards:
Organizing committee
ICAC2N – 2024

Registration Confirmation 1st IEEE ICAC2N-2024 : Paper ID 658 @ ITS Engineering College, Greater Noida

1 message

Microsoft CMT <email@msr-cmt.org> Thu, Jun 13, 2024 at 6:34 PM
Reply-To: "Dr. Vishnu Sharma" <vishnu.sharma@its.edu.in>
To: Sanchit Agarwal <sanchitagarwal969@gmail.com>

Dear Sanchit Agarwal,
Delhi Technological University,New Delhi

Greetings from ICAC2N-2024 ...!!! Thanks for Completing your registration...!!

Paper ID- "658 "
Paper Title- " A survey of various Machine Learning and Deep Learning Approaches in the field of Information Security "

This email is to confirm that you have successfully completed your registration for your accepted paper at ICAC2N-2024. We have received your registration and payment details. Further, your submitted documents will be checked minutely and if any action will be required at your end you will be informed separately via email.

For further updated regarding conference please keep visiting conference website www.icac2n.in or write us at icac2n.ieee@gmail.com.

Regards:
Organizing committee
ICAC2N – 2024

Note:
1. Transfer of E-copyright to IEEE and Presenting paper in conference is compulsory for publication of paper in IEEE. (For this you will be informed separately via email well before conference)
2. If plagiarism is found at any stage in your accepted paper, the registration will be cancelled and paper will be rejected and the authors will be responsible for any consequences. Plagiarism must be less then 15% (checked through Turnitin).
3. Change in paper title, name of authors or affiliation of authors is not allowed now.
4. Violation of any of the above point may lead to cancellation of registration.
5. Registration fee once paid is non-refundable.

Gmail Sanchit Agarwal <sanchitagarwal969@gmail.com>

Acceptance Mail-ICAAIML 2024

1 message

iccse VGNT <iccse@vignanits.ac.in> Tue, May 28, 2024 at 7:50 PM
To: Sanchit Agarwal <sanchitagarwal969@gmail.com>

Dear Sanchit

It is our pleasure to inform you that your papers entitled **Evaluating LSTM and Bi-LSTM for Binary Classification in Intrusion Detection System in IoT** (Paper Id: ICAAIML -21) has been provisionally accepted for Virtual oral paper presentation at ICAAIML-2024 on 30th and 31st August 2024, and also your paper has been accepted to publish in **AIP conference proceeding (SCOPUS)**

We request you to complete the early bird conference registration fee and publication charges i,e Rs 3000+ publication charges Rs 8500= 11,500 **If you don't want AIP publication then just pay Rs 3000 only**, After payment send the payment proof along with full manuscript.

Pay the registration fee through

Bank A/C No 00421140047879

Account Name : B Sridhar Babu

Bank Name: HDFC

IFSC Code: HDFC0000042

For conference updates please join our telegram channel : https://t.me/+VioSEzuF3b5NSzJ4

Thank you

with regards

ICAAIML

Gmail Sanchit Agarwal <sanchitagarwal969@gmail.com>

Registration-- Reg

1 message

iccse VGNT <iccse@vignanits.ac.in> Thu, Jun 13, 2024 at 7:45 PM
To: Sanchit Agarwal <sanchitagarwal969@gmail.com>

Dear Author
we received your payment, Thankyou for registration, please join telegram channel for conference updates, if you have queries please mail us
thank you

PAPER NAME
Sanchit Thesis.pdf

AUTHOR
Sanchit Agarwal

WORD COUNT
9804 Words

CHARACTER COUNT
53682 Characters

PAGE COUNT
42 Pages

FILE SIZE
642.6KB

SUBMISSION DATE
May 29, 2024 1:45 PM GMT+5:30

REPORT DATE
May 29, 2024 1:46 PM GMT+5:30

10% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 6% Internet database
- 5% Publications database
- Crossref database
- Crossref Posted Content database
- 7% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Small Matches (Less then 10 words)

Sanchit Agarwal

Sanchit Thesis.pdf

My Files

My Files

Delhi Technological University

Document Details

Submission ID

trn:oid:::27535:59392809

Submission Date

May 17, 2024, 1:40 AM GMT+5:30

Download Date

May 17, 2024, 1:44 AM GMT+5:30

File Name

Sanchit Thesis without starting pages.pdf

File Size

642.6 KB

42 Pages

9,804 Words

53,682 Characters

How much of this submission has been generated by AI?

0%

of qualifying text in this submission has been determined to be generated by AI.

Caution: Percentage may not indicate academic misconduct. Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

What does the percentage mean?
The percentage shown in the AI writing detection indicator and in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was generated by AI.

Our testing has found that there is a higher incidence of false positives when the percentage is less than 20. In order to reduce the likelihood of misinterpretation, the AI indicator will display an asterisk for percentages less than 20 to call attention to the fact that the score is less reliable.

However, the final decision on whether any misconduct has occurred rests with the reviewer/instructor. They should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in greater detail according to their school's policies.

How does Turnitin's indicator address false positives?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be AI-generated will be highlighted blue on the submission text.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

What does 'qualifying text' mean?
Sometimes false positives (incorrectly flagging human-written text as AI-generated), can include lists without a lot of structural variation, text that literally repeats itself, or text that has been paraphrased without developing new ideas. If our indicator shows a higher amount of AI writing in such text, we advise you to take that into consideration when looking at the percentage indicated.

In a longer document with a mix of authentic writing and AI generated text, it can be difficult to exactly determine where the AI writing begins and original writing ends, but our model should give you a reliable guide to start conversations with the submitting student.