A Real-Time Deep Neural Network-Based System's Ability to Identify and Classify Active Assaults on Networks

Karthikeyan Kaliyaperumal [1,*], Raja Sarath Kumar Boddu [2], and Sai Kiran Oruganti [3,*]

[1] School of Global Postdoctoral and Research, Lincoln University College, Pedaling, Jaya, Malaysia
[2] Department of Computer Science and Engineering, Raghu Engineering College, Visakhapatnam, India
[3] Faculty of Engineering and Built Science, Lincoln University College, Pedaling, Jaya, Malaysia
Email: pdf.kirithicraj@lincoln.edu.my (K.K.); rajaboddu@lincoln.edu.my (R.S.K.B.); saisharma@lincoln.edu.my (S.K.O.)
*Corresponding author

***Abstract*—In the context of computer science and technology, a group of linked devices, or nodes, that exchange data, resources, or services with one another is referred to as a network. An active network attack refers to a malicious activity in which an attacker deliberately attempts to disrupt, manipulate, or gain unauthorized access to a computer network or its resources. In today's digital context, network security is of vital importance as cyber threats continue to evolve in terms of sophistication and frequency. Active network attacks pose significant challenges to traditional detection methods, necessitating the exploration of advanced techniques such as deep learning. This research proposes a novel approach for the identification and classification of active network attacks based on deep learning methodologies. To achieve this, an experimental research analysis design was used. A comprehensive review of deep learning approaches appropriate for network attack detection was undertaken. The proposed methodology involves the development of a model based on deep learning that was learned using a dataset comprising diverse network traffic data which is Network Security Laboratory-Knowledge Discovery in Databases (NSL-KDD). This study utilizes a comprehensive preprocessing pipeline, including data cleaning, feature selection for categorical variables and standardization of numerical features to prepare the dataset for modeling. To extract the pertinent information, preprocessing approaches are used. Metrics like as accuracy, precision, recall, F1-Score, and confusion matrix are used to evaluate performance as a result from deep learning models Deep Neural Network (DNN), Convolution Neural Networks (CNN), Long Short Term Memory(LSTM), Bi-Long Short Term Memory (Bi-LSTM) and Gated Recurrent Units (GRU) experiments done, Bi-LSTM model scored the best result of 99.15% and 99.12% accuracy for binary and multi classification, respectively.**

***Keywords*—active network, cyber attacks, Deep Neural Network (DNN), Convolution Neural Networks (CNN), deep learning, detection, neural networks**

I. Introduction

Active network attack detection and classification using deep neural network learning model involve using deep learning techniques to identify and classify various types of network attacks in real time or near real time. Deep learning has shown promise in many fields, including cyber security. Building and training deep learning model using huge network dataset, researchers and practitioners can potentially create more effective and robust intrusion detection systems [1]. The term "active network attacks" typically refers to malicious activities aimed at disrupting or gaining unauthorized access to computer networks. Detecting and classifying these attacks is crucial for maintaining the security and integrity of networked systems [2].

Using deep learning for this purpose offers several advantages. Deep learning models can automatically extract relevant features from raw network data, eliminating the need for feature engineering. They can also adapt and learn from new attack patterns, making them potentially more proficient at detecting previously unseen threats. Additionally, deep learning models can handle with large-scale datasets efficiently, enabling the detection of attacks in real time or near real time [3].

In the context of computer science, a network refers to a collection of two or more computers linked or connected together to share the resources, exchange files, or enable electronic via communications [4]. It is networked nodes and entities that share information with one another. it plays a fundamental role in modern computing, enabling communication between devices, sharing resources, and facilitating various services and applications [5].

A. Statement of the Problem

Modern society has developed because of computer networks, which have an impact on public services, economic growth, healthcare, education, social development, and innovation. By facilitating effective communication, information accessibility, and the smooth functioning of diverse industries, networks play a vital role in the general advancement and well-being of individuals and communities [6].

However, as information technology advances, network security has become a new issue. Numerous network security events arise as a result of the network's growing

Manuscript received May 6, 2025; revised June 21, 2025; accepted July 8, 2025; published November 7, 2025.

size and continuously growing number of network users [7].

According to Ref. [8], more than 20,000 cyber-attacks have resulted in direct losses for financial firms of billion since 2020. The chief director general clarified that were cyber-attack attempts succeeded, distributed denial-of-service for a predetermined amount of time, lost revenue, prevented operations, stolen data, hidden or encrypted data, claimed money to retrieve, and fraudulently used communication channels would have happened to generate income illegally. According to the Director General, financial institutions, security institutions, media outlets, important government offices, ministries, regional offices, hospitals, and higher education establishments made up the majority of the targets of cyber-attacks.

To solve related problems stated above, a number of studies have been performed using traditional and advanced machine learning methods globally. Ieracitano *et al.* [9] employed NSL-KDD dataset in order to implement the method of intelligent intrusion detection driven by auto encoders. Even though it is encouraging, improvements are still needed to increase its accuracy and dependability.

Other researchers have also performed deep learning-based cyber-attack detection for the Internet of Medical Things (IoMT) [10]. The approach achieved a significant accuracy of 96.39%, but it was limited to man-in-the-middle attacks, indicating the need for a more comprehensive detection system that can handle a wider range of cyber threats.

Anwer *et al.* [11] have also employed classical algorithms which are Gradient Boosted Decision Tree (GBDT), and Random Forest (RF) algorithms to detect attacks in networks. The random forest algorithm scored 85.34% of accuracy that is relatively low in the context of securing network environments, suggesting the necessity for more effective models.

Since traditional machine learning methods have difficulty in efficiently detecting and classifying attackers because cyber security issues take the form of new and sophisticated methods, it is essential to research and develop novel methods using deep learning techniques.

The difficulty in choosing features as the amount of data increases is reducing the attack detection rate in terms of traditional machine learning. Remote to Local (R2L), User to Root (U2R), probe, and Denial of Service (DoS) attack types classified with low accuracy.

B. High False Positive Rates

Moreover, due to the challenges in feature selection and classification accuracy, there is a risk of high false positive rates, where benign activities are incorrectly classified as attacks. Overall, these issues can significantly impact the effectiveness of attack detection systems, potentially leading to an increased risk of security breaches and false alarms.

C. Objectives

The general objective of this study is to develop a deep learning model for difficulty in efficiently detecting and classifying active network attack.

II. Literature Review

A. Overview of Network

There are several types of networks, including Local Area Networks (LANs), which cover a specific area, such as a building, and Wide Area Networks (WANs), which connect several LANs across larger territories. Wireless networks, such as Wi-Fi and cellular networks, transmit data via radio or infrared signals. Peer-to-Peer (P2P) networks enable direct device communication without the need for a central server, which is often used for file sharing. Client-server networks, on the other hand, allow clients to access services from centralized servers, which is common in online applications and cloud computing. Network components include nodes (devices), connections (communication channels), protocols (data transmission regulations), and infrastructure (hardware and software support) [5].

B. Network Security and Attack Detection

The Internet's widespread use and continuous expansion are beneficial to many network users in many ways and preventing unwanted access and alteration is the goal of defense when it comes to computers, networks, programs, different types of data. This is where network security comes in [12].

However, as more and more systems in the financial, e-commerce, and military become internet-connected, they become targets for network attacks, which increases risk and causes significant harm. In essence, effective tactics for attack detection, defense, and network security maintenance are required. Thus, the primary problem in the field of network security to be solved and how to recognize various types of network attacks [13].

C. Machine Learning in Network Security

Traditional models have been widely employed in attack detection and classification in network security. However, Traditional machine learning approaches rely heavily on handcrafted feature engineering, where domain experts manually design and select relevant features from raw network traffic data. These features may include packet headers, flow statistics, payload content, or behavior-based features extracted from network logs. Feature engineering aims to capture discriminative information that distinguishes between normal network behavior and malicious activities [14].

1) Supervised learning

Supervised learning algorithms, such as decision trees, random forests, Support Vector Machines (SVMs), K-Nearest Neighbors (KNNs), and logistic regression, are trained on labeled datasets to identify the network attack traffic instances as benign or malicious. These classifiers learn decision boundaries or decision rules based on input features an enabling them to distinguish between various kinds of attacks and normal network behavior [15].

2) Machine learning versus deep learning

A lot of Machine Learning (ML) is used to recognize different kinds of attacks. A machine learning methodology could help the network administrator take the necessary steps to prevent breaches. However, the

majorities of conventional machine learning techniques fall within the category of shallow learning and often pay attention to feature selection and engineering. shallow learning is unable to effectively address the categorization problem when faced with massive amounts of intrusion huge data that emerge in a real-time network platform [5].

In contrast, deep learning techniques can generate considerably more effective prototypes and have the ability to derive better representations from dynamic data sets [16].

A collection of techniques known as "representation learning" or "feature learning" in classical models makes the algorithm automatically learn the representations needed for feature detection from the training dataset. On the contrary, Deep Learning (DL) may be viewed as establishing machine learning and representation learning jointly. With many levels of cumulative complexity and generalization, as well as the final prediction, DL aims to jointly learn fundamental traits. The key distinction between Deep Learning (DL) and Machine Learning (ML) is depicted in the Fig. 1. DL uses automated feature selection while standard ML uses manual feature selection.

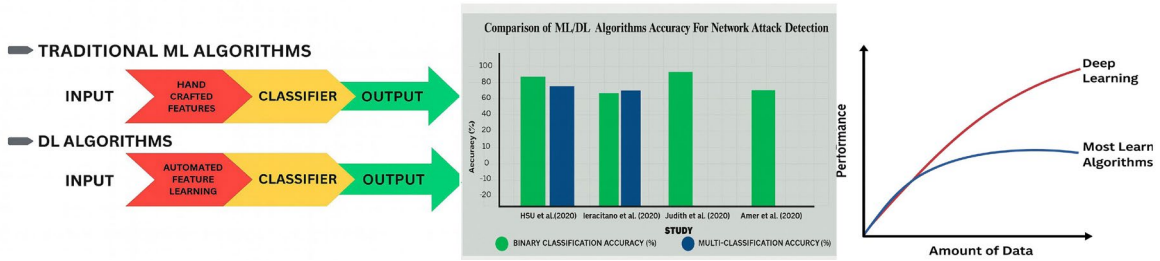


Fig. 1. Deep learning versus machine learning.

3) Limitations of machine learning

Manual Feature Engineering: Traditional machine learning model approaches often rely on feature engineering. Handcrafted features may fail to capture refined or complex patterns in the data and may not generalize well to new scenarios [17].

Limited Generalization: Traditional machine learning models may struggle to generalize effectively to new attack patterns, especially in dynamic and evolving network environments. Models trained on historical data may become outdated or ineffective in detecting novel or sophisticated attack strategies, which leads toward reduced detection performance [13].

Limited Adaptability: Traditional machine learning models may have limited adaptability to changing network conditions, attack tactics, and enemy strategies over time. These models may not dynamically adjust to new attack patterns or concept drift in network traffic data, requiring frequent retraining or manual intervention to maintain effectiveness.

4) Deep learning approaches

Deep learning has emerged as a powerful approach for active network attack detection and classification, offering significant advantages over traditional methods [18].

a) Feature learning

Deep learning models can automatically learn hierarchical representations of raw input data, such as network traffic packets or logs, without the need for handcrafted feature engineering. By processing raw data through multiple layers of nonlinear transformations, deep learning models extract informative features directly from the input, enabling them to capture complex patterns and relationships that may be difficult to specify manually.

b) Real-time feature engineering

Feature engineering has its roots in the batch processing paradigm. This approach involved collecting large volumes of historical data, processing it offline in scheduled intervals, and using the results to train models. The lag between data generation and feature availability.

The landscape began to shift as businesses realized the immense value locked in real-time data streams. Real-time feature engineering emerged as a response to this need, allowing for the continuous computation of features as data arrives. This approach enables models to operate on the most current data, dramatically reducing latency between event occurrence and model reaction.

5) Deep Neural Networks (DNN)

DNNs can be employed for general network attack detection by learning patterns in network traffic data. It consists of multiple layers of neurons, typically including an input layer, several hidden layers, and an output layer. Each layer performs linear combinations of inputs followed by a non-linear activation function. DNNs can be fully connected; meaning every neuron in one layer is connected to every neuron in the subsequent layer. DNNs are versatile and can learn complex patterns from network traffic data. Their ability to learn from large datasets makes them effective for identifying previously unseen attack patterns [19].

6) Convolutional Neural Networks (CNN)

CNNs are commonly used for analyzing spatial data, but they can also be applied to sequential data, such as network traffic sequences. In the context of active attack detection, CNNs can learn spatial features from network traffic data, such as packet headers or content, to identify characteristic patterns associated with different types of attacks. By twisting filters across input sequences and combining spatial information, CNNs can effectively capture local dependencies and spatial correlations in network traffic data, enabling accurate detection and classification of active attacks [20].

7) Long Short Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a unique kind of artificial Recurrent Neural Network (RNN) architecture

that is utilized in the deep learning field. It is effective in detecting network attacks due to their ability to capture long-term dependencies in sequential data, particularly in analyzing time-series data like traffic logs or sequences. For example, LSTMs can identify anomalies in user behavior by analyzing sequences of actions taken over time, helping to distinguish between legitimate and malicious activities. They consist of memory cells and information-controlling gates (input, forget, and output gates). Because of this, LSTMs may retain and pick up dependencies throughout lengthy sequences. LSTMs are ideally suited for studying sequential data, such as traffic flows over time, which makes them useful for network attack detection [21].

8) Bidirectional LSTMs (Bi-LSTMs)

Bi-LSTMs extend the capabilities of LSTMs by processing data in both forward and backward directions. This bidirectional approach allows the model to consider context from both past and future states, enhancing its ability to detect complex attack patterns. Bi-LSTMs have been shown to outperform traditional LSTM models in tasks requiring a deeper understanding of context, such as identifying specific types of network attacks based on historical traffic behavior [22].

9) Gated Recurrent Units (GRUs)

GRUs are similar to the LSTMs with a simplified architecture that combines the forget and input gates into a single update gate. This makes GRUs computationally efficient while still effectively capturing dependencies in sequential data. In network attack detection, GRUs can be used to analyze patterns in network traffics.

D. Types of Network Attacks

Broadly applicable security attacks are classified into passive attacks and active attacks. A passive attack attempts to learn or make use of information from the system but does not affect system resources, whereas an active attack attempts to alter system resources or affect their operation. Any effort to alter the system without authorization is considered an active attack. For instance, this could involve altering data that have been sent or stored, generating new data streams through masquerading or fabrication, replaying or changing messages, and causing a denial of service or availability disruption [8].

Network attack detection involves the proactive monitoring of network traffic, system logs, and behavior patterns to quickly identify and respond to unauthorized access attempts, malware infections, and other forms of cyber-attacks [23]. Our research will focus on active network attack detection and classification. Focusing on active network attack detection and classification is vital because of increasing sophistication as well as prevalence of network attacks. Active attacks, where hackers attempt to alter or disrupt network operations, pose significant risks to the integrity and availability of systems. In this study, attack types are classified using the following network attack classes:

Denial of Service (DoS): A DoS attack aims to overwhelm a system or network resource, making it unavailable to the users. This type of attack disrupts services by flooding the target with excessive traffic or requests, causing it to crash or become unresponsive [18].

Remote-to-Local (R2L): R2L attacks involve unauthorized users attempting to connect remotely and obtain local access to a system. Attackers exploit vulnerabilities of a system to increase their privileges and gain unauthorized access to sensitive data or resources [18].

User-to-Root (U2R): U2R attacks involve as users with limited privileges attempting to benefit of the administrative access to a system. Attackers exploit vulnerabilities to increase their privileges and gain unauthorized control over the system, potentially leading to data breaches or system compromise [23].

Probe: Probe attacks involve attackers scanning a network to gather necessary things about potential vulnerabilities and system configurations. These attacks are reconnaissance activities aimed at identifying weaknesses that could be exploited in subsequent attacks [24].

E. Relevance of Works

Active network attacks involve attackers actively launching attacks against target servers, where the attacker attempts to change the data on the target. These attacks can include unauthorized changes to the system, such as the alteration of transmitting data and stored as well, the fabrication of data, masquerade attacks, messages replays, messages modifications, including service denial attacks [25]. These network components need to be reliable and secured through advanced deep learning technologies to detect and mitigate anomalies provided a comprehensive survey of Intrusion Detection Systems (IDSs) tailored for Wireless Sensor Networks (WSNs) [26]. Their work classified IDS based on detection approaches and deployment strategies and laid a foundational framework for understanding the landscape of intrusion.

Building upon this taxonomy, presented a review focused on machine learning techniques for network intrusion detection. By synthesizing advancements in machine learning algorithms and their application to intrusion detection, the authors highlight the potential of these techniques in enhancing the accuracy and efficiency of network defense mechanisms.

A comprehensive study of deep learning techniques for anomaly detection was carried out. Deep learning techniques have emerged as potential approaches for anomaly detection in network traffic [27]. Demonstrating the effectiveness of neural network architectures in capturing complicated patterns indicative of malicious activities. In a similar investigated the application of deep learning approaches specifically for network intrusion detection [28]. Their review offers insights into the design and evaluation of deep learning models, emphasizing their scalability and adaptability to evolving threat landscapes.

Furthermore, Tun *et al.* [29] offer an overview of network anomaly detection techniques, emphasizing the importance of a comprehensive classification to categorize detection methods based on their objectives and methodologies. Their work provides a holistic perspective

on the diverse range of approaches employed in the detection and classification of network attacks.

Several studies [30−32] have highlighted the limitations of traditional misuse detection methods, such as signature-based Intrusion Detection Systems (IDSs). These methods rely on known attack signatures and struggle to detect novel or zero-day attacks, leading to increased vulnerability to emerging threats [33]. Hsu *et al.* [34] have emphasized the potential benefits of hybrid approaches that combine multiple detection methods to improve detection accuracy and resilience against evolving threats. By integrating misuse and anomaly detection methods using deep learning, it is possible to leverage the complementary strengths of both approaches and achieve more robust and accurate detection outcomes. Collectively, these studies contribute to advancing the state of the art in active network attack detection and classification. By synthesizing insights from various domains, including wireless sensor networks, machine learning, deep learning, security, researchers and practitioners are empowered to develop more resilient and adaptive intrusion detection systems capable of mitigating emerging threats in dynamic network environments.

Therefore, deep learning techniques can offer a data-driven and adaptive approach to active attack detection and classification, enabling the development of more accurate, efficient, and robust security systems capable of defending against evolving cyber threats in complex network environments. A novel approach to intelligent intrusion detection using auto encoder-driven intelligence and statistical analysis was developed by researchers, which achieved 87% accuracy for malt classification and 84.21% accuracy for binary classification using NSL-KDD [9]. Even though the work is appreciated, it still needs more improvement. Other studies have also explored the use of Long Short-Term Memory (LSTM)-based convolutional neural networks to detect network intrusions. They emphasize the growing relevance of network security as the internet becomes more widely used. Researchers have suggested two deep learning models, LSTM-only and CNN-LSTM, to increase the performance of intrusion detection systems, with the NSL-KDD dataset serving as a benchmark. This work aimed to solve the constraints of existing machine learning algorithms in intrusion detection, and it achieved 94.12% and 88.95% accuracy for binary classification and multi-classification, respectively [35].

Judith *et al.* [10] has performed work to detect cyber-security threats, with a particular focus on man-in-the-middle attacks that occur within the Internet of Medical Things (IoMT) communication network. Principle Component Analysis (PCA) has been utilized for feature reduction and employs a multilayer perceptron to classify unforeseen cyber-attack IoT-based healthcare devices. The study results indicated that the multilayer perceptron outperforms the other tested classifiers, achieving an accuracy of 96.39% while also improving the performance by reducing the time complexity. Even if the accuracy is significant, it is particularly focused on only man-in-the-middle attacks. Sharafaldin *et al.* [36] has also performed work to identify IoT attacks using machine learning algorithms, namely, Support Vector Machines (SVMs), Gradient Boosted Decision Trees (GBDTs), and Random Forests (RFs), with RF-based supervised machine learning algorithms achieving an accuracy of 85.34%. The study scores low accuracy in the context of a secure network.

Compared intrusion detection systems, specifically examining Apriority, which use data mining association rule techniques, and Support Vector Machine, which utilizes machine learning methodologies [37]. We evaluate the two systems using the UNSW-NB15 and NSL-KDD datasets, which were developed by the University of New South Wales for research in Knowledge Discovery and Data Mining. The Scientific and Technical Network (STN) sector used several machine learning methods to pre-empt suspect traffic. Employing Support Vector Machine (SVM) based Kernel Principal Component Analysis (KPCA), the dimensionality of feature vectors was reduced, and Genetic Algorithms (GA) were used to optimize different SVM parameters. An enhanced kernel function (N-RBF) was used to mitigate noise resulting from feature discrepancies. The experimental results indicated that the model surpassed a singular SVM regarding generalization and classification accuracy.

F. Analysis of Dataset Axioms

CICIDS2017 Dataset: Modern Attack Scenarios: CICIDS2017 includes a wide range of modern attack scenarios, such as DoS, DDoS, Brute Force, and Web attacks, making it more representative of current cyber threats.

Realistic Network Traffic: The dataset contains realistic network traffic, including both normal and malicious activities, which helps in training more effective intrusion detection models.

Labeled Data: CICIDS2017 is labeled, making it suitable for supervised learning approaches.

UNSW-NB15 Dataset: Comprehensive Attack Coverage: UNSW-NB15 includes a comprehensive set of attack types, such as Fuzzers, Analysis, Backdoors, and Exploits, providing a more realistic representation of modern cyber threats. Real-World Network Traffic: The dataset is generated from real-world network traffic, making it more relevant for training intrusion detection models. Hybrid Attack Scenarios: UNSW-NB15 includes hybrid attack scenarios, which are more challenging to detect and require more sophisticated detection models.

G. Relevance to Deep Learning

Complex Patterns: Both CICIDS2017 and UNSW-NB15 datasets contain complex patterns and relationships, making them suitable for deep learning-based approaches. Large-Scale Data: These datasets are large-scale, which is essential for training deep learning models that require significant amounts of data to learn effectively. In summary, the CICIDS2017 and UNSW-NB15 datasets are more relevant and modern, providing a more accurate representation of current cyber threats and realistic network traffic. They are well-suited for training deep learning-based intrusion detection models.

The RNN achieved an accuracy of 99% in binary categorization. The objective of the deep learning model created by Yang *et al.* [6] was to detect malicious traffic inside an encrypted network. The proposed model originated from a Residual Neural Network (ResNet). The adversarial sample of encrypted traffic was produced with Deep Convolution Generative Adversarial Networks (DCGAN) and Deep Q-Network (DQN) reinforcement learning. The problem of uneven and inadequate samples was solved. The accuracy of the model was 99.94%. indicating exceptional performance. To achieving a False Acceptance Rate (FAR) of 2.94%) and a False Rejection Rate (FRR) of 2.28%.) A technique for the early identification of distributed Denial-of-Service (DDoS) assaults executed via a botnet integrates real network data with deep Convolutional Neural Networks (CNNs) [9].

To execute a coordinated distributed Denial-of-Service (DDoS) attack inside a cell that might impair CPS operations, the puppet device oscillates between quiet calls, Simple Mail Services (SMS) spamming, or a combination of these tactics aimed at disrupting calls, Internet access, SMS, signaling, or a blend of primarily focused on an intrusion detection system using a hybrid placement strategy that integrates multi-agent systems, blockchain technology, and deep learning algorithms [7].

Collectively, these studies contribute to advancing the state of the art in active network attack detection and classification. By synthesizing insights from various domains, including, machine learning, deep learning, security, researchers and practitioners are empowered to develop more resilient and adaptive intrusion detection systems capable of mitigating emerging threats in dynamic network environments.

Therefore, deep learning techniques can offer a data-driven and adaptive approach to active attack detection and classification, enabling the development of more accurate, efficient, and robust security systems capable of defending against evolving cyber threats in complex network environments.

A novel approach to intelligent intrusion detection using auto encoder-driven intelligence and statistical analysis was developed by researchers, which achieved 87% accuracy for malt classification and 84.21% accuracy for binary classification using NSL-KDD [9]. Even though the work is appreciated, it still needs more improvement. Other studies have also explored the use of Long Short-Term Memory (LSTM)-based convolutional neural networks to detect network intrusions.

They emphasize the growing relevance of network security as the internet becomes more widely used [35]. Researchers have suggested two deep learning models, LSTM-only and CNN-LSTM, to increase the performance of intrusion detection systems, with the NSL-KDD dataset serving as a benchmark. This work aimed to solve the constraints of existing machine learning algorithms in intrusion detection, and it achieved 94.12% and 88.95% accuracy for binary classification and multi-classification, respectively [38].

A study entitled Efficient Deep Learning-Based Cyber-Attack Detection for internet of Medical Things Device has also been performed to detect cyber-security threats [10]. PCA has been utilized for feature reduction and employs a multilayer perceptron to classify unforeseen cyber-attack healthcare devices.

This study results indicated that the multilayer perceptron, achieving an accuracy of 96.39% complexity [10]. Even if the accuracy is significant, it is particularly focused on only man-in-the-middle attacks. The summary of related works as shown in the Table I.

TABLE I. SUMMARY OF RELATED WORKS

S/No	Ref.	Title	Technique	Accuracy	Gaps
1	[19]	(LSTM)-based CNN to detect network intrusions	Deep Learning	94.12% and 88.95% for binary classification and multi-classification, respectively.	- Compares only two models - Accuracy needs to be improved in both case
2	[21]	A Novel Statistical Analysis and Auto encoder Driven Intelligent Intrusion Detection Approach	Deep Learning	84.21% and 87%. For Binary (0's and 1's) classification and Multi-classification, respectively	- Done for both Binary classification and Multi-classification however, - Low Accuracy for both case
3	[25]	Efficient Deep Learning-Based Cyber-Attack Detection for IoMT devices.	Deep Learning	96.39%	- Focused on only man-in-the- middle attack
4	[9]	Attack Detection in IoT using Machine Learning algorithms	Machine Learning	85.34%	- Low accuracy - Used traditional machine learning which cannot detect novel anomalies without manually upgraded

III. RESEARCH METHODOLOGY AND DESIGN

A. Introduction

The methodology is a method for systematically and scientifically solving a research problem. The most popular techniques are experimentation, observation, surveys, interviews, focus groups, and archival research. Data collection and analysis procedures are known as research methods. Surveys, experiments, interviews, and observations are examples of common methodologies [39]. Consequently, to conduct this research, we used an experimental research methodology.

Because, experimental research provides a quantitative basis for evaluating the performance of detection systems in terms of accuracy, detection rate, false positive rate, response time, and other relevant metrics. This allows researchers to compare different approaches and identify the most effective solutions for detecting and justifying specific types of network attacks. This section describes

and explains the critical research methods for detecting and classifying active network attacks.

B. Dataset Used

The availability of effective data set is a requirement for an improvement of intelligent attack detection system. An attack detection system can only benefit training and testing with a dataset which contains a lot of high quality data and simulates a crucial time. In the recent field of computer network attack detection, the NSL-KDD dataset is a commonly recognized benchmark dataset. This dataset is a carefully designed addressing the shortcomings of the original NSL-KDD Cup 99 dataset, offering an improved and more accurate version of the latter. The NSL KDD dataset contains a wide range of network traffic data, including both normal and various types of attack activity. It thus becomes a precious tool for the assessment and development of network attack detection systems. Its capacity to accurately simulate real-world network traffic conditions increases its applicability, making it a top option for practitioners and researchers looking to systematically evaluate and improve their network attack detection algorithms [40].

They stated that researchers can access this publicly available data source, which offers the following benefits over the original KDD data set: It does not include redundant records in the train set, so the classifiers will not be biased towards more frequent records. There are no duplicate records in the proposed test sets; therefore, the performance of the learners is not biased by the methods which have better detection rates on the frequent records. The number of selected records from each difficulty level group is inversely proportional to the percentage of records in the original KDD data set. As a result, the classification rates of distinct machine learning methods vary in a wider range, which makes it more efficient to have an accurate evaluation of different learning techniques.

Furthermore, the number of records in the train and test sets is reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research works will be consistent and comparable [41].

All the above stated points are why we selected the dataset for this study. The dataset each record contains 41 attributes that describe different aspects of the flow and are categorized as either an attack type or normal. Table II include information about the qualities, including their name, descriptions, and sample data.

TABLE II. FEATURES PROVIDED BY NSL-KDD

No	Name of feature	Description	Type
1.	Duration	Connection duration (TCP, UDP, ICMP and the like) in seconds	Numeric
2.	Protocal_type	Connection protocol	Nominal
3.	Service	Dst port mapped to service (HTTP, FTP, etc.)	Nominal
4.	Flag	Normal or error status flag of connection	Nominal
5.	Src_bytes	Number of data bytes from src to dst	Numeric
6.	Dst_bytes	Bytes from dst to src	Numeric
7.	Land	1 if connection is from /to the same host/port: else 0	Binary
8.	Wrong_fragment	Number of 'wrong' fragment (e.g., 0,1,3)	Numeric
9.	Urgent	Number of urgent packets	Numeric
10.	Hot	Number of 'hot' indicators (bio-ids feature)	Numeric
11.	Num_failed_login	Number of failed login attempts	Numeric
12.	Logged_in	1if successfully logged in; else 0	Binary
13.	Num_compromized	Number of 'compromised' conditions	Numeric
14.	Root_shell	1 if root shell is obtained; else 0	Binary
15.	Su_attemped	1 if 'su root' command attempted; else 0	Binary
16.	Num_root	Number of root access	Numeric
17.	Num_file_creation	Number of file creation operation	Numeric
18.	Num_shells	Number of shell prompts	Numeric
19.	Num_access_files	Number of operations on access control files	Numeric
20.	Num_outbound_cmds	Numbers of outbound commands in an FTP session	Numeric
21.	Is_host_login	1 if login belongs to 'hot' list (e.g., Root, admin; else 0	Binary
22.	Is_guest_login	1 if login 'guest' login (e.g., guest, anonymous); else 0	Binary
23.	Count	Number of connections to the same host as a current connection in the past two seconds	Numeric
24.	Srv_count	Number of connections to the same service as a current connection in the past two seconds	Numeric
25.	Serror_rate	% of connections that have 'SYN' Errors	Numeric
26.	Srv_serror_rate	% of connections that have 'SYN' Errors	Numeric
27.	Rerror_rate	The connection that has 'REJ' Errors	Numeric
28.	Srv_serror_rate	The connection that has 'REJ' Errors	Numeric
29.	Same_srv_rate	Connections that have the same services	Numeric
30.	Diff_srv_rate	Connections to different services	Numeric
31	Srv_diff_host_rate	Connections to different hosts	Numeric
32.	Dst_host_count	Count of connections having the same dst host	Numeric
33.	Dst_host_srv_count	Count of connections having the same host and using the same service	Numeric
34.	Dst_host_same_srv_rate	Connections having the same host dst and using the same service	Numeric
35.	Dst_host_diff_srv_rate	Different services on current host	Numeric
36.	Dst_host_same_srv_por t_rate	Connections to current host having same srv port	Numeric
37.	Dst_host_srv_diff_host _rate	Connection to same service coming from different hosts	Number
38.	Dst_host_serror_rate	Connections to the current host that have an S0 error	Numeric
39.	Dst_host_srv_serror_rate	Connections to current host and specified service that have an S0 error	Numeric
40.	Dst_host_rerror_rate	Connections to current host that have an RST error	Numeric
41.	Dst_host_srv_rerror_rate	Connections to the current host and specified services that have an RST error	Numeric

C. Data Preprocessing

Once the dataset is collected, it needs to be preprocessed. The data should be cleaned, displayed, and subjected to feature engineering. This could involve tasks such as normalization, dealing with missing values, feature selection and analysis of the data.

The data preprocessing is crucial in our analysis, as redundant features can lead to confusion in the training process of machine learning models. This redundancy can negatively impact the performance of these models, particularly when optimized using techniques like Bayesian Optimization for hyperparameter tuning. By removing redundant features and handling missing values appropriately, we can improve the efficiency and accuracy of our models.

1) Data cleaning

The practice of eliminating duplicate, erroneous, improperly formatted, malformed, or corrupt data from a dataset is known as data cleaning. There is a great chance that data will be mislabeled or duplicated when it is combined from many sources. Inaccurate data can produce untrustworthy conclusions and algorithms, even in cases where the data is accurate. Because the techniques will differ based on the dataset, there is no one-size-fits-all way to prescribe the exact stages of the data cleaning process.

2) String indexing

Nominal, numeric, and binary feature data as included in the dataset that was used to create the suggested anomaly detection and classification model. Because of this, we need to convert the data from classification to numeric or from a text to the correct format before we can run a model on it. The String Indexer transformer is used to convert categorical text data into numerical data that can be modeled. Based on how frequently a category value appears in the input columns of a data frame, the String Indexer turns it into an index. The input column will be converted from numeric to string before counting if it is numeric. While the label indicates whether a sample is normal or an attack, the feature in this case is the entire row that is assigned to columns or fields.

D. Performance Measurement Methods

1) Loss function

It is an approach to assess how effectively our machine learning algorithm predicts the data set. In other words, loss functions are a measure can predict the desired result. For this study, we used Binary Cross-Entropy and Categorical cross entropy loss functions.

2) Binary cross-entropy

Binary Cross-Entropy (BCE) is a loss function commonly used in binary classification tasks, where the goal is to classify data into one of two possible categories (e.g., attack or no attack in Network Attack Detection model) [42].

The formula for Binary Cross-Entropy is:

$$BCE = -\frac{1}{N}\sum_{i=1}^{N}[yi\ log(y\hat{}i) + (1 - yi)\ log(1 - y\hat{}i)]$$

where:

- yi is the actual label (0 or 1).
- $y\hat{}i$ is the predicted probability for the positive class (i.e., 1).
- N is the number of samples.

The Binary Cross-Entropy loss penalizes wrong predictions, especially when the predicted probability is far from the actual label. A model that minimizes this loss will produce predictions that closely match the true labels. It's particularly well-suited for tasks where the output is a probability between 0 and 1 (such as when using a sigmoid activation function in the output layer of a neural network). But, for multi classification we used categorical cross entropy.

E. Model Performance Evaluation

Model's performance on test dataset indicates at what extent it will perform in the real world. The performance of the models in this study is evaluated using five evaluation metrics based on the literature review: confusions matrix, accuracy, precisions, recalls, and F1-score.

F. Evaluation Techniques

The next step after the model has been trained is to access the model effectiveness and success rate using various metrics. The fundamental performance indicators we utilized for the network attack detection and classification model are listed below: recall, F1-score, confusion matrix, precision, accuracy.

G. Model Architectural

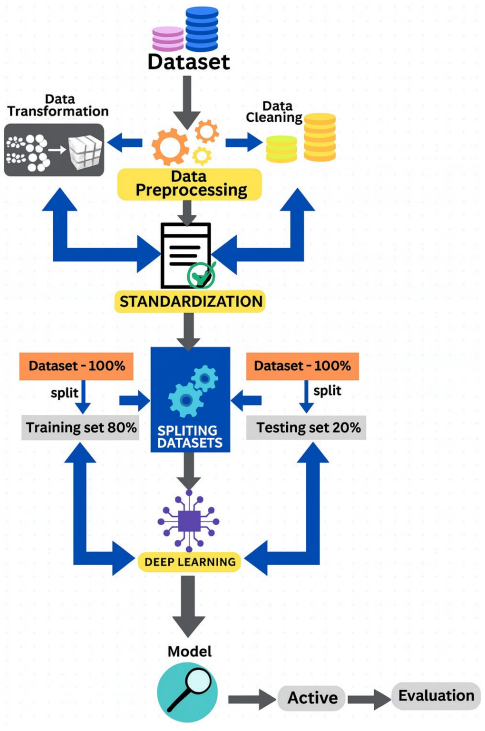


Fig. 2. Architecture evaluation model.

The proposed deep learning models used for active network attack detection and classification were trained and tested using NSL-KDD dataset for both binary and

multi classifications. Because, forty-one characteristics, including connection duration, protocol type, service type, and amount of bytes transferred, are included in the NSL-KDD dataset and describe different aspects of network traffic. By balancing the data, the majority class is less biased and models are trained more successfully. For scientific research, the NSL-KDD dataset is easily accessible. This shows all of the processing steps seen in Fig. 2.

H. Description of the Model

Five hidden layers make up the Deep Neural Network (DNN), and each layer has 128, 64, 32, 16, and 8 neurons. All hidden layers employ the Rectified Linear Unit (ReLU) activation function. To enhance generalization and reduce overfitting, a dropout layer with a rate of 0.2 follows each hidden layer. The Methods of Regularization: Adam optimizer with a learning rate of 0.001 and L2 regularization with a penalty term of 0.01.

Three convolutional layers with 32, 64, and 128 filters in each layer make up a Convolutional Neural Network (CNN). Each convolutional layer uses the ReLU activation function. After every convolutional and dropout layer, a max-pooling operation with a pool size of 2×2 is applied. Following the pooling layers, two fully connected layers with 128 and 64 neurons, respectively, are used, each followed by a dropout layer with a dropout rate of 0.2. The network is optimized using the Adam optimizer with a learning rate of 0.001.

Network of Long Short-Term Memory (LSTM): There are two LSTM layers and 128–64 units in each layer. The hyperbolic tangent (tanh) activation function is used for both the dropout and LSTM layers. A dropout rate of 0.2 is applied after each LSTM layer to prevent overfitting. L2 regularization is employed during training, using the Adam optimizer with a learning rate of 0.001 and a regularization penalty term of 0.01.

There are two Bi-LSTM layers and 128–64 units in each layer. The hyperbolic tangent (tanh) activation function is employed for both the dropout and Bi-LSTM layers. A dropout rate of 0.2 is applied after each Bi-LSTM layer to prevent overfitting. The Methods of Regularization: L2 regularization using the Adam optimizer with a learning rate of 0.001 and a penalty term of 0.01.

There are two GRU layers, and each layer contains 128, 64 units. The hyperbolic tangent (tanh) and sigmoid activation functions are employed in the dropout and GRU layers. A dropout rate of 0.2 is applied after each GRU layer to prevent overfitting. The Methods of Regularization: L2 regularization using the Adam optimizer with a learning rate of 0.001 and a penalty term of 0.01. By changing the number of layers, units, and hyperparameter functions, they can be customized to fit certain issues and datasets.

IV. Results and Discussions

A. Overviews

The experiment conducted to assess the effectiveness of the suggested models is covered in this chapter. In this process`s different tools utilized such as libraries, datasets, implementation specifics, and performance evaluation outcomes of models utilizing various evaluation criteria are all addressed.

B. Dataset Used

In this study, we utilized NSL-KDD dataset which is refined version of predecessor KDD99. Because, it addresses issue of redundant record in that found in its predecessor dataset. As stated above in Section III, the original dataset contained many duplicate records, which could lead to biased training results. NSL-KDD eliminates this redundancy.

The NSL-KDD dataset also includes a wide variety of attack types, categorized into four major classes which are DoS: Attempts to make a machine or network resource unavailable to its intended users. Probe: Attempts to gather information about a network or system, often as a precursor to an attack. R2L: Attempts to gain unauthorized access to a machine over a network. U2R: Attempts to gain unauthorized root privileges on a machine. This diversity makes the dataset suitable for training models that can detect and classify different types of network attacks.

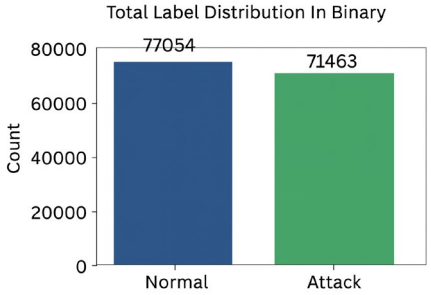


Fig. 3. Binary classification in training set and test set.

TABLE III. Binary Classification in Train and Test

Category	Train	Test	Total
Normal	61,643	15,411	77,054
Dos	42,708	10,677	53,385
R2L	2999	750	3749
U2R	202	50	252
Probe	11,261	2816	14,077
Total	118,813	29,704	148,517

Note: Set using 80% and 20% dataset split.

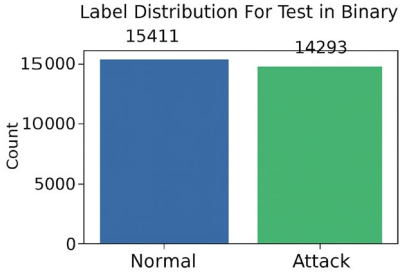


Fig. 4. Binary classification label distribution for test dataset.

As shown in Fig. 3 and Table III, in the binary classification results from the total dataset, 77,054 instances were classified as normal, whereas 71,463 instances were classified as attack. From these, 15,411 normal and 14,293 attack samples were used for testing, as illustrated in Fig. 4.

C. Implementation Details

In this study we were explained how the implementation is going on. The first steps are acquiring the datasets and the data is preprocessed then string indexer is applied to in both train dataset and test datasets which convert the string data to the numeric which are 0 for normal and 1 for attack.

1) Network attack in terms of binary classification

The NSL-KDD dataset which was used in this experiment classified into train set and test set. The training set is used to train the classification algorithm, while the test set consists of new instances that are not present in the training data and are used to evaluate the model's performance.

The total number of data which incorporated in this experiment is 148,517 records the dataset utilized were divided into normal and attack categories. In this study, 80% of data dataset is used for the training and 20% of data for testing. With this dataset binary classification and multi-class classification were trained and tested.

2) Network Attack in terms of multi classification

NSL-KDD was used in experiments which dived into two categories. It is train category and test category. We used 80% and 20% of total dataset for training and testing respectively, while test data is used make up of create new instance which is not found in the train data and use to create a model. The number of data used in NSL-KDD train and NSL-KDD test were shown in Table IV below.

TABLE IV. MULTI CLASSIFICATION IN TRAIN AND TEST USING 80 % AND 20% DATASET SPLIT

Category	Train	Test	Total
Normal	61,643	15,411	77,054
Attack	57,170	14,293	71,463
Total	118,813	29,704	148,517

The total number of data used in these experiments was 148,517 with train accounting for 80% and test accounting for 20%. There are 118,813 instances in the train data. (Normal is 61,643, Dos attack 42,708, R2L attack is 2999, U2R attack is 202, and Probe attack is 11,261) where as there are 29,704 instances in test data which are (Normal 15,411, Dos attack 10,677, R2L attack 750, U2R attack 50 attack Probe attack 2816) in test dataset split.

Normal and Dos have large instance across datasets used for testing and training. As indicated in the Table IV, next to probe, R2L and U2R have small instances in both train and test dataset. From the dataset we split into five labels which means normal labels 77,054 Dos 53,385, Probe 14,077, R2L 3749 and U2R 252 network is more affected by Dos and Probe respectively. From these we used for test 15,411, 10,677, 2816, 750, 50 normal, DoS Probe, R2L, U2R as it is shown on the following Figs. 5 and 6.

The following are the class or category that attacks in train and test instances mapped to:

DOS: Appache, back, land, mail, bomb, netpune, pod, processtable, smurf, teardrop, udpstorm.

R2L: ftp-write, Guess password, Internet Message Access Protocol (IMAP), mult hop, named, phf, send mail, Snmpget attack, Snmp guess, spy, warez, client, warez master, worm, X lock, X snoop.

U2R: Buffer overflow, laodmodule, perl, rootkit, httptunnel, ps, sqlattack and xterm.

Probe: IP sweep, Mscan, Namp, Port sweep, saint.

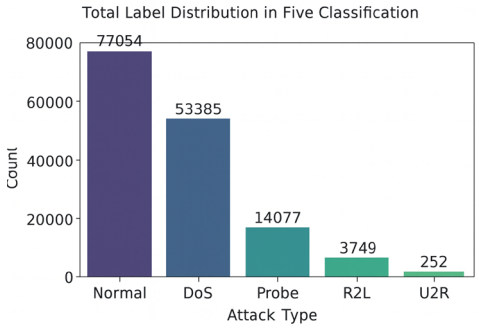


Fig. 5. Total distribution of normal and attack types.

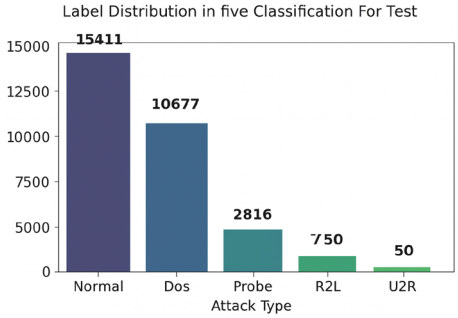


Fig. 6. Distribution of normal and attack types for test.

D. Hyper Parameters Used for all Models in this Research

An epoch is one complete pass of the training dataset in the algorithm. This epoch number is important hyper parameters for the algorithm. It shows how many full iterations or the whole set of training data undergoing during the algorithm training or processing. Training will come to conclusion at the time number of iterations surpasses number of epochs. When trained by minimum error, with the maximum number of iterations. Batch size is the number of samples that pass through to the network at one time. Table V illustrates that 50 epochs were used for binary classification and 100 epochs for multi-class classification, both with a batch size of 512 and optimized using a Bayesian optimization algorithm.

TABLE V. HYPER PARAMETERS USED IN BOTH BINARY AND MULTI CLASSIFICATION

Models	Epochs		Batch Size
	For binary classification	For multi classification	
DNN	50	100	512
CNN	50	100	512
LSTM	50	100	512
Bi-LSTM	50	100	512
GRU	50	100	512

Fig. 7 shows the classification report of the performance classification outcome for DNN utilizing the binary classes' a confusion matrix the counts of TP, TN, FP, and

TN for each class are displayed in the above classification confusion matrix normal (0) and attack (1). For each class the model produced results which are true positive and true negative out of the normal test samples (15,411), 15,292 are correctly categorized as normal, 119 normal samples are mistakenly categorized as attack and from 14,293 attack samples, 14,135 are correctly classified as attack whereas 158 instances incorrectly classified as normal.

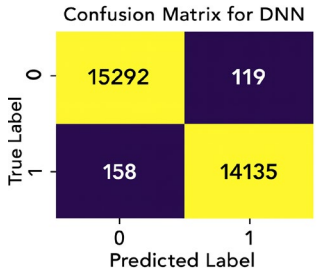


Fig. 7. Confusion matrix in DNN for binary classification.

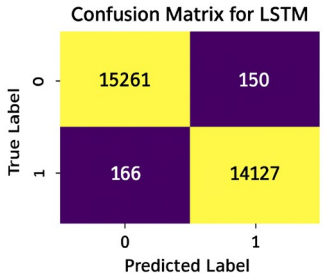


Fig. 8. Confusion matrix in LSTM for binary classification.

TABLE VI. CLASSIFICATION REPORT FOR ALL MODELS

Models	Classification	Precision	Recall	F1-Score	Support
DNN	0	0.99	0.99	0.99	15,411
	1	0.99	0.99	0.99	14,293
	Accuracy	-	-	0.99	29,704
	Macro avg	0.99	0.99	0.99	29,704
	Weighted avg	0.99	0.99	0.99	29,704
CNN	0	0.99	0.99	0.99	15,411
	1	0.99	0.99	0.99	14,293
	Accuracy	-	-	0.99	29,704
	Macro avg	0.99	0.99	0.99	29,704
	Weighted avg	0.99	0.99	0.99	29,704
Bi-LSTM	0	0.99	0.99	0.99	15,411
	1	0.99	0.99	0.99	14,293
	Accuracy	-	-	0.99	29,704
	Macro avg	0.99	0.99	0.99	29,704
	Weighted avg	0.99	0.99	0.99	29,704
GRU	0	0.99	0.99	0.99	15,411
	1	0.99	0.99	0.99	14,293
	Accuracy	-	-	0.99	29,704
	Macro avg	0.99	0.99	0.99	29,704
	Weighted avg	0.99	0.99	0.99	29,704

As shown in Fig. 8, the performance report of the LSTM model for binary classification is illustrated using confusion matrices, which display the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for the normal (0) and attack (1) classes. Out of 15,411 normal test samples, 15,261 were correctly classified as normal, while 150 were misclassified as attacks. Similarly, among 14,293 attack samples, 14,127 were correctly identified as attacks, whereas 166 were incorrectly classified as normal.

Table VI shows the performance classification report outcome for all models (DNN, CNN, Bi-LSTM, GRU) utilizing the binary classes' confusion matrices the counts of TP, TN, FP, and TN for every class normal (0) and attack (1).

E. Classification Report and Confusion Matrix in CNN for Binary Classification

As shown in Fig. 9, the performance of the CNN model for binary classification is presented using confusion matrices, which display the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for the normal (0) and attack (1) classes. Out of 15,411 normal test samples, 15,200 were correctly classified as normal, while 211 were misclassified as attacks. Similarly, among 14,293 attack samples, 14,180 were correctly identified as attacks, whereas 113 were incorrectly classified as normal.

For each class the model produced results which are true positive and true negative out of the normal test samples (15,411), 15,278 are correctly categorized as normal, 133 normal samples are mistakenly categorized as attack and from 14,293 attack samples, 14,173 are correctly classified as attack whereas 120 instances incorrectly classified as normal as in shown in Fig. 10.

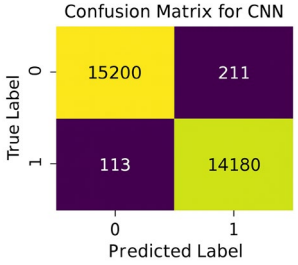


Fig. 9. Confusion matrix for CNN classification.

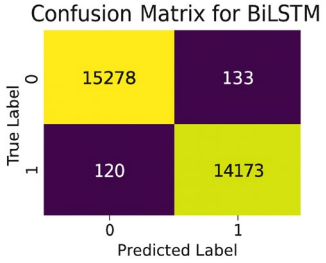


Fig. 10. Confusion matrix in Bi-LSTM in terms of binary classification.

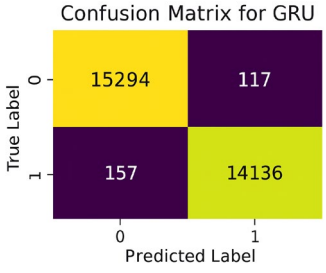


Fig. 11. Confusion matrix in GRU for binary classification.

As shown in Fig. 11, the model produced the following binary classification results: out of 15,411 normal test samples, 15,294 were correctly classified as normal, while 117 were misclassified as attacks. Similarly, among 14,293 attack samples, 14,136 were correctly identified as attacks, whereas 157 were incorrectly classified as normal.

Among models used in this study, Bi-LSTM scored 99.23% validation accuracy during training while LSTM, GRU, DNN and CNN scored 99.21%, 99.17%, 99.16%, and 99.07% respectively for binary classification as shown in Fig. 12.

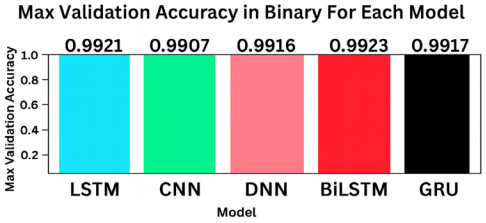


Fig. 12. Max validation accuracy in binary for each model.

F. *Classification Report and Confusion Matrix in DNN for Multi Classification*

TABLE VII. CLASSIFICATION REPORT FOR ALL MODELS

Models	Classification	Precision	Recall	F1-Score	Support
DNN	DoS	1.00	1.00	1.00	10,677
	Probe	0.99	0.97	0.98	2816
	R2L	0.88	0.93	0.90	750
	U2R	0.90	0.70	0.79	50
	Normal	0.99	0.99	0.99	15,411
	Accuracy	-	-	0.99	29,704
	Macro avg	0.95	0.92	0.93	29,704
	Weighted avg	0.99	0.99	0.99	29,704
CNN	DoS	0.99	1.00	1.00	10,677
	Probe	0.99	0.96	0.97	2816
	R2L	0.90	0.74	0.81	750
	U2R	0.78	0.62	0.69	50
	Normal	0.98	0.99	0.99	15,411
	Accuracy	-	-	0.98	29,704
	Macro avg	0.93	0.86	0.89	29,704
	Weighted avg	0.98	0.98	0.98	29,704
LSTM	DoS	1.00	1.00	1.00	10,677
	Probe	0.98	0.98	0.98	2816
	R2L	0.87	0.89	0.88	750
	U2R	0.88	0.74	0.80	50
	Normal	0.99	0.99	0.99	15,411
	Accuracy	-	-	0.99	29,704
	Macro avg	0.95	0.92	0.93	29,704
	Weighted avg	0.99	0.99	0.99	29,704
BI-LSTM	DoS	1.00	1.00	1.00	10,677
	Probe	0.98	0.99	0.99	2816
	R2L	0.90	0.91	0.91	750
	U2R	0.78	0.72	0.75	50
	Normal	0.99	0.99	0.99	15,411
	Accuracy	-	-	0.99	29,704
	Macro avg	0.93	0.92	0.93	29,704
	Weighted avg	0.99	0.99	0.99	29,704
GRU	DoS	1.00	1.00	1.00	10,677
	Probe	0.99	0.98	0.99	2816
	R2L	0.90	0.92	0.91	750
	U2R	0.84	0.74	0.79	50
	Normal	0.99	0.99	0.99	15,411
	Accuracy	-	-	0.99	29,704
	Macro avg	0.94	0.93	0.93	29,704
	Weighted avg	0.99	0.99	0.99	29,704

The performance classification outcome for DNN utilizing the five classes' confusion matrices the TP, TN, FP, and TN counts for each class (Dos, probe, R2L, U2R, and normal) can be found in Table VII. For each class the model produced results which are true positive and true negative. Out of the DoS test samples (10,677), 10,651 are correctly classified as DoS whereas none (0), 5, 0, 21 of DoS samples are incorrectly classified as probe, R2L, U2R, and normal, respectively.

Among the total 2,816 Probe samples, 2,744 were correctly classified as Probe, while 4, 1, 1, and 66 Probe samples were incorrectly classified as DoS, R2L, U2R, and Normal, respectively. For the R2L class, out of 750 test samples, 697 were correctly identified as R2L, whereas 1, 0, 2, and 50 samples were misclassified as DoS, Probe, U2R, and Normal, respectively.

Among the total U2R (50) sample, 35 samples are correctly classified as U2R whereas 0, 0, 3, 12 U2R sample are incorrectly classified as Dos, Probe, R2L, and normal, respectively. In the normal class from test sample (15,411), 15,291 sample are correctly classified as normal whereas 6, 27, 86, 1 of normal sample are incorrectly classified as Dos, Probe, R2L, and U2R, respectively.

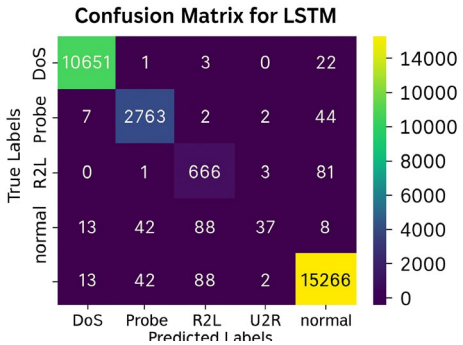


Fig. 13. Confusion matrix in CNN for Multi classification.

Table VII shows the performance classification conclusion for using the five classes' confusion matrices for each class (Dos, probe, R2L, U2R, and normal).

For each class the model produced results which are true positive and true negative. Out of the DoS test samples (10,677), 10,652 are correctly classified as DoS whereas 3, none (0), 2, 20 of DoS samples are incorrectly classified as probe, R2L, U2R, and normal, respectively.

Among the total 2,816 Probe samples, 2,707 were correctly classified as Probe, whereas 15, 1, 1, 92 Probe sample are incorrectly classified as Dos, R2L, U2R, and normal, respectively. In the R2L class from test sample (750), 555 sample correctly classified as R2L whereas 1, 0, 1, 193 samples of R2L wrongly classified as DoS, Probe, U2R, and normal, respectively.

Among the total 50 U2R samples, 31 were correctly classified as U2R. whereas 2, 0, 3, 14 U2R sample are incorrectly classified as Dos, Probe, R2L, and normal, respectively. In the normal class from test sample (15,411), 15,275 sample are correctly classified as normal whereas 38, 37, 56, 5 of normal sample are incorrectly classified as Dos, Probe, R2L, and U2R, respectively, as shown in the Fig. 13.

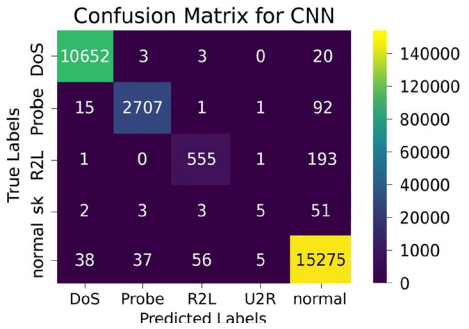


Fig. 14. Confusion matrix in LSTM for multi classification.

For each class the model produced results which are true positive and true negative. Out of the DoS test samples (10,677), 10,651 are correctly classified as DoS whereas 1, 3, none (0), 22 of DoS samples are incorrectly classified as probe, R2L, U2R, and normal, respectively.

Among the total Probe (2816) sample, 2763 sample are correctly classified as Probe whereas 7, 2, 0, 44 Probe sample are incorrectly classified as Dos, R2L, U2R, and normal, respectively. In the R2L class from test sample (750), 666 sample correctly classified as R2L whereas 0, 0, 3, 81 samples of R2L wrongly classified as DoS, Probe, U2R, and normal, respectively.

Among the total U2R (50) sample, 37 sample are correctly classified as U2R whereas 2, 0, 3, 8 U2R sample are incorrectly classified as Dos, Probe, R2L, and normal, respectively. In the normal class from test sample (15,411), 15,266 sample are correctly classified as normal whereas 13, 42, 88, 2 of normal sample are incorrectly classified as Dos, Probe, R2L, and U2R, respectively, as shown in the Fig. 14.

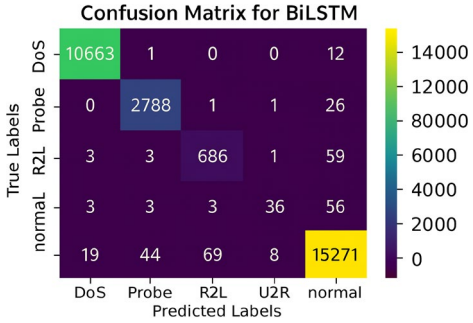


Fig. 15. Confusion matrix in Bi-LSTM for multi classification.

The performance classification outcome for Bi-LSTM utilizing the five classes' confusion matrices the TP, TN, FP, and TN counts for each class (Dos, probe, R2L, U2R, and normal) are shown in Table VII. For each class the model produced results which are true positive and true negative. Out of the DoS test samples (10,677), 10,663 are correctly classified as DoS whereas 1, 0, 0, 13 of DoS samples are incorrectly classified as probe, R2L, U2R, and normal, respectively.

Among the total Probe (2816) sample, 2788 sample are correctly classified as Probe whereas 0, 1, 0, 26 Probe sample are incorrectly classified as Dos, R2L, U2R, and normal, respectively. In the R2L class from test sample (750), 686 sample correctly classified as R2L whereas 1, 3, 1, 59 samples of R2L wrongly classified as DoS, Probe, U2R, and normal, respectively.

Among the total U2R (50) sample, 36 sample are correctly classified as U2R whereas 0, 1, 3, 10 U2R sample are incorrectly classified as Dos, Probe, R2L, and normal, respectively. In the normal class from test sample (15,411), 15,271 sample are correctly classified as normal whereas 19, 44, 69, 8 of normal sample are incorrectly classified as Dos, Probe, R2L, and U2R, respectively, as shown in the Fig. 15.

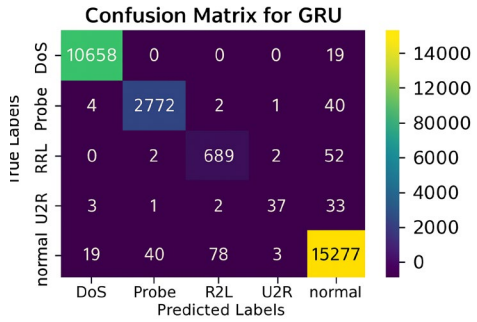


Fig. 16. Confusion matrix in GRU for multi classification.

Finally, the performance classification outcome for GRU utilizing the five classes' confusion matrices the TP, TN, FP, and TN counts for each class (Dos, probe, R2L, U2R, and normal) can be seen in Table VII. For each class the model produced results which are true positive and true negative. Out of the DoS test samples (10,677), 10,658 are correctly classified as DoS whereas 0, 0, 0, 19 of DoS samples are incorrectly classified as probe, R2L, U2R, and normal, respectively.

Among the total Probe (2816) sample, 2772 sample are correctly classified as Probe whereas 3, 0, 1, 40 Probe sample are incorrectly classified as Dos, R2L, U2R, and normal, respectively.

In the R2L class from test sample (750), 689 sample correctly classified as R2L whereas 4, 2, 3, 52 samples of R2L wrongly classified as DoS, Probe, U2R, and normal, respectively.

Among the total U2R (50) sample, 37 sample are correctly classified as U2R whereas 2, 1, 2, 8 U2R sample are incorrectly classified as Dos, Probe, R2L, and normal, respectively. In the normal class from test sample (15,411), 15,277 sample are correctly classified as normal whereas 23, 30, 78, 3 of normal sample are incorrectly classified as Dos, Probe, R2L, and U2R, respectively, as shown in the Fig. 16.

Among models used in this study, DNN scored 99.16% validation accuracy during training while Bi-LSTM, GRU, LSTM and CNN scored 99.15%, 99.11%, 99.11%, and 98.54%, respectively, for binary and multi classification. In this case, the Deep Neural Network (DNN) model demonstrated superior performance compared to all other models.

As shown in Fig. 17, which illustrates the training and validation progress of the proposed DNN model, the training accuracy starts at approximately 94%, while the validation accuracy begins at around 97%.

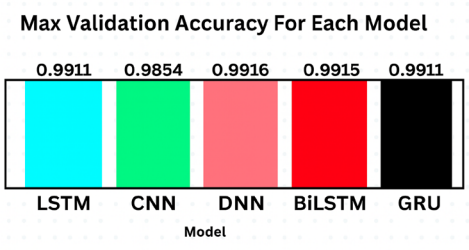
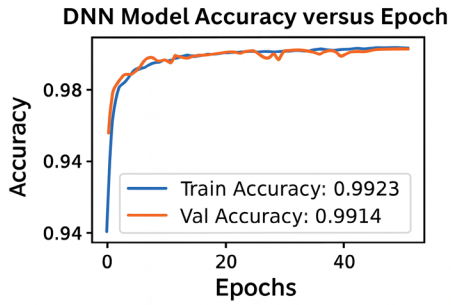


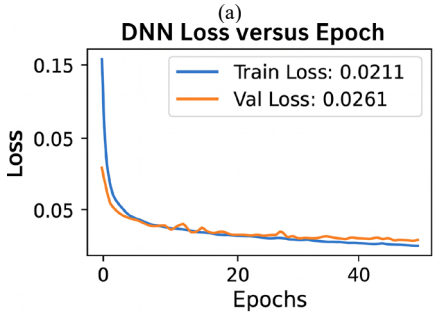
Fig. 17. Max validation accuracy for each model in multi classification.

The validation accuracy value rapidly increases until the 50th epoch, surpassing the values of the training accuracy line as it began to climb with decreases and increase until the final epoch. When it both reach 99.23% and around 99.14% during the 50th epoch, there is 0.005% gaps between training accuracy and validation accuracy.

The training and validation loss curves start at 0.15 and 0.09, respectively. Thereafter, the validation loss increases almost linearly, and the training loss shows a corresponding upward trend.



(a)



(b)

Fig. 18. Accuracy versus loss with their corresponding epochs for binary classification using DNN.

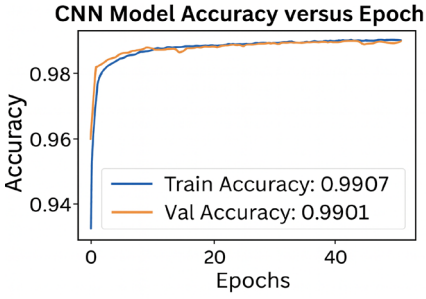
According to Fig. 18, which shows the train and validation progress of the suggested DNN model, the training accuracy value line begins nearly 93% while the validation accuracy value begins at nearly 95.5%.

The validation accuracy value rapidly increases until the 50th epoch, surpassing and overlapping the values of the training accuracy line as it began to climb with decreases and increase until the final epoch.

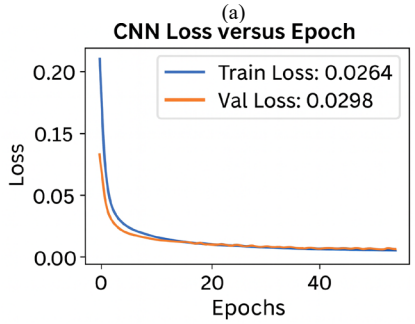
When it both reach 99.07% and around 99.01% during the 50th epoch, there is 0.003% gaps between training accuracy and validation accuracy.

The start point of training and validation loss curves are mentioned as 0.20 and 0.13, respectively. The validation loss value goes down then experience nearly straight decrease and training loss value also decrease.

Fig. 19(a) and (b) illustrate the train and validation progress of the proposed CNN model. The training accuracy value line starts at around 91%, while the validation accuracy value starts at approximately 96%. The training accuracy line started to climb with reductions and increase until the final epoch, and the validation accuracy value occasionally surpassed and fell below the values of the training accuracy line as it rapidly increased and decreased until the 50th epoch.



(a)



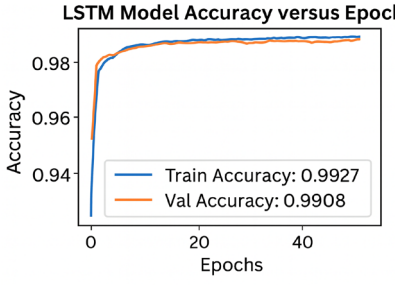
(b)

Fig. 19. Accuracy versus loss with their corresponding epochs for binary classification using CNN.

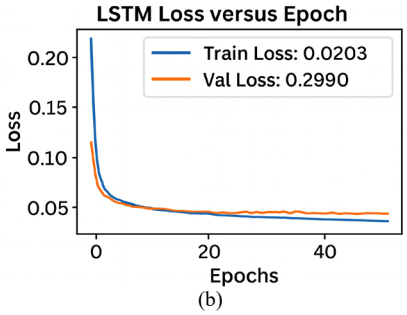
When it both reach 99.27% and around 99.08% during the 50th epoch, there is 0.008% gaps between training accuracy and validation accuracy. The start point of training and validation loss curves are shown in Fig. 19 as 0.21 and 0.10, respectively.

The validation loss value goes down then experience nearly straight increase and training loss value also increase.

As Fig. 20 shows, the train and validation progress of the suggested LSTM model, the training accuracy value line begins nearly 92% while the validation accuracy value begins at nearly 97%. The validation accuracy value rapidly increases until the 50th epoch, surpassing the values of the training accuracy line as it began to climb with decreases and increase until the final epoch.



(a)

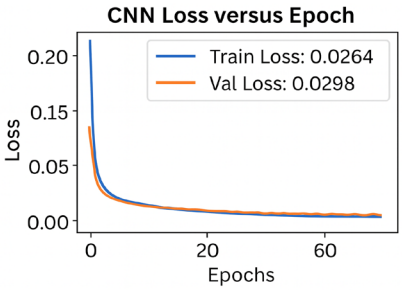


(b)

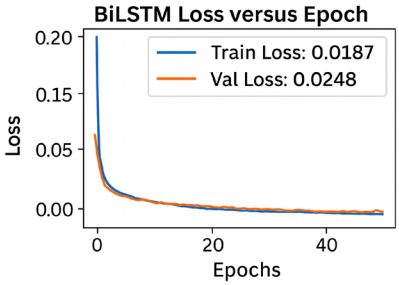
Fig. 20. Accuracy versus loss with their corresponding epochs for binary classification using LSTM.

When it both reach 99.31% and around 99.21% during the 50th epoch, there is 0.006% gaps between training accuracy and validation accuracy. The start point of training and validation loss curves are shown in Fig. 20 as 0.20 and 0.10, respectively. The validation loss value goes down then experience nearly straight decrease and training loss value also decrease.

According to Fig. 21(a) and (b) shows which the train and validation progress of the suggested Bi-LSTM model, the training accuracy value line begins nearly 91% while the validation accuracy value begins at nearly 96%. The validation accuracy value rapidly increases until the 50th epoch, surpassing the values of the training accuracy line as it began to climb with decreases and increase until the final epoch.



(a)



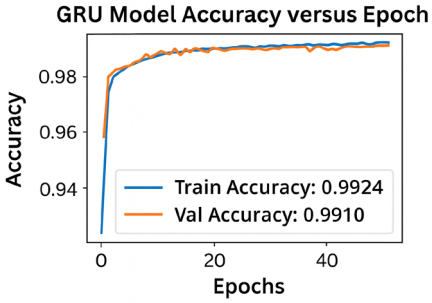
(b)

Fig. 21. Accuracy versus loss with their corresponding epochs for binary classification using Bi-LSTM.

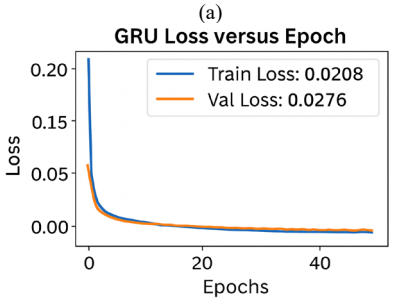
When it both reach 99.24% and around 99.10% during the 50th epoch, there is 0.006% gaps between training accuracy and validation accuracy. The start point of training and validation loss curves are shown in Fig. 21 (right) as 0.20 and 0.10, respectively. The validation loss value goes down then experience nearly straight decrease and training loss value also decrease.

When we sum up these above analyses according to the following, DNN and GRU training accuracy is above and its validation loss is below all other models, respectively.

According to Fig. 22(a) and (b) which shows the train and validation progress of the suggested GRU model, the training accuracy value line begins nearly 85% while the validation accuracy value begins at nearly 90%. The validation accuracy value rapidly increases then when reaches above 99.38% it decreases until some epochs and again increases and finally increased but below training accuracy until the final epoch.



(a)



(b)

Fig. 22. Accuracy versus loss with their corresponding epochs for binary classification using GRU.

When it both reach 99.38% and around 99.02% during the 1000th epoch, there is 0.002% gaps between training accuracy and validation accuracy. The start point of training and validation loss curves are shown in Fig. 22 (right) as 0.5 and 0.23, respectively. The validation loss value goes down then experience goes up increase again decrease and training loss value also in the same way.

According to Fig. 23(a) and (b) shows which the train and validation progress of the suggested DNN model, the training accuracy value line begins nearly 92% while the validation accuracy value begins at nearly 96%. The validation accuracy value rapidly increases then when reaches above 98% decreases until 100th epoch, being below the values of the training accuracy line until the final epoch.

When it both reach 99.37% and around 98.98% during the 100th epoch, there is 0.002% gaps between training accuracy and validation accuracy. The start point of training and validation loss curves are shown in Fig. 23 is 0.3 and 0.1, respectively. The validation loss value then goes down experience nearly straight increase and training loss value also increases again decrease.

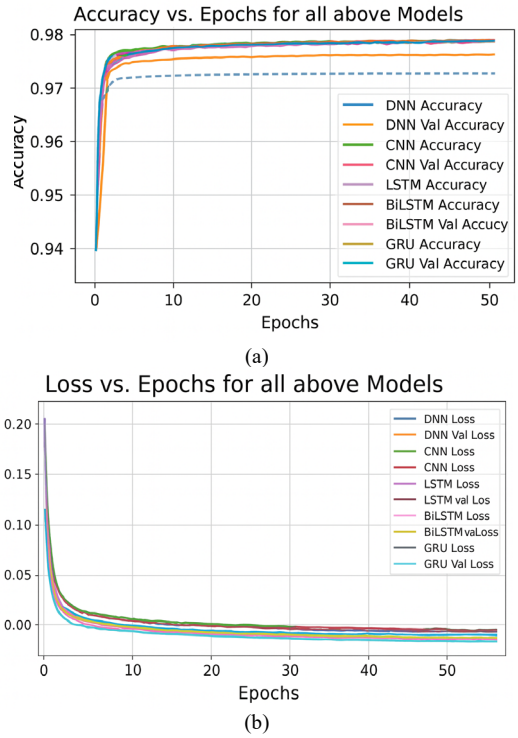


Fig. 23. Loss versus epochs for all above models using binary classification.

As shown in Fig. 24(a) and (b), which the train and validation progress of the suggested DNN model, the training accuracy value line begins nearly 80% while the validation accuracy value begins at nearly 90%. The validation accuracy value rapidly increases then when reaches above 98% almost it continues as straight line until 100th epoch, showing nearly same values of the training accuracy line until the final epoch.

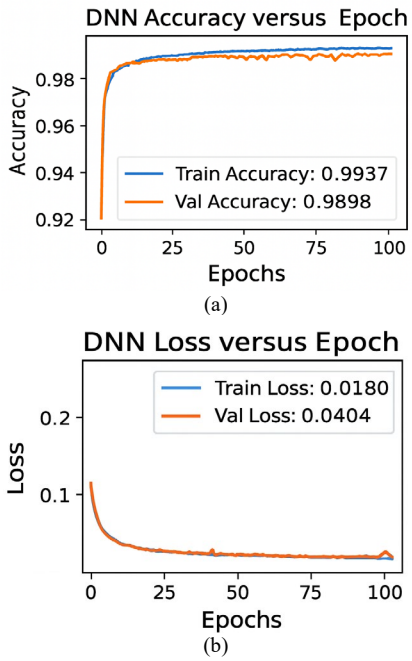


Fig. 24. Accuracy versus loss with their corresponding epochs for multi classification using DNN.

When it both reach 98.49% and around 98.25% during the 100th epoch, there is 0.004% gaps between training accuracy and validation accuracy. The start point of training and validation loss curves are shown in Fig. 24 as 0.7 and 0.2, respectively. The validation loss value goes down then experience nearly straight decrease and training loss value also decrease.

According to Fig. 25(a) and (b) which shows the train and validation progress of the suggested model, the training accuracy value line begins nearly 80% while the validation accuracy value begins at nearly 92%. The validation accuracy value rapidly increases then when reaches above 98% almost it continues as straight line until 100th epoch, showing nearly same values of the training accuracy line until the final epoch. When it both reach 99.31% and around 98.88% during the 100th epoch, there is 0.002 % gaps between training accuracy and validation accuracy.

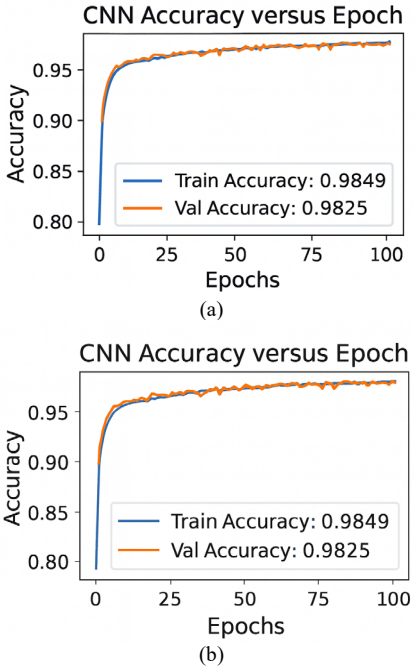


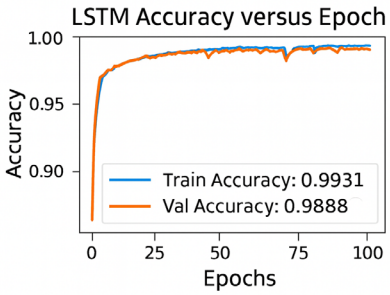
Fig. 25. Accuracy versus loss with their corresponding epochs for multi classification using CNN.

The start point of training and validation loss curves are shown in Fig. 25 as 0.5 and 0.2, respectively. The validation loss value goes down then experience nearly up and down and training loss value also in same way.

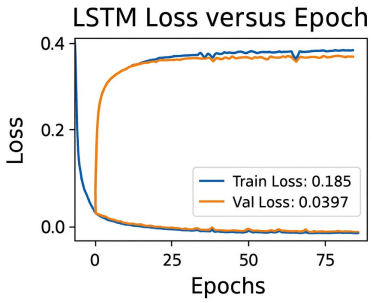
According to Fig. 26(a) and (b) shows, which the train and validation progress of the suggested Bi-LSTM model, the training accuracy value line begins nearly 80% while the validation accuracy value begins at nearly 93%. The validation accuracy value rapidly increases then when reaches above 99% almost it continues as straight line but below training accuracy until 100th epoch.

When it both reach 99.34% and around 99.12% during the 100th epoch, there's 0.002% gaps between training accuracy and validation accuracy. The start point of training and validation loss curves are shown in Fig. 26 as 0.4 and 0.2, respectively. The validation loss value goes

down then experience nearly straight decrease and training loss value also decrease.



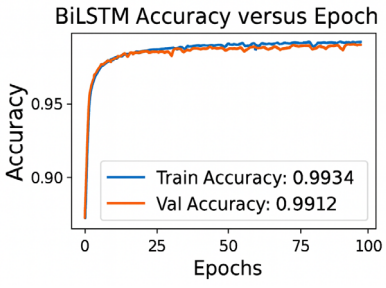
(a)



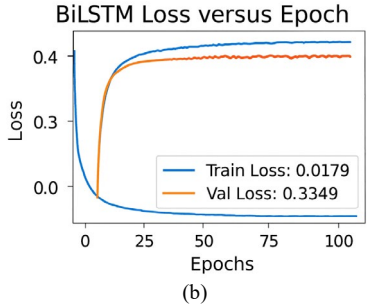
(b)

Fig. 26. Accuracy versus loss with their corresponding epochs for multi classification using LSTM.

According to Fig. 27(a) and (b) which shows the train and validation progress of the suggested GRU model, the training accuracy value line begins nearly 85% while the validation accuracy value begins at nearly 90%. The validation accuracy value rapidly increases then when reaches above 99.34% it decreases until some epochs and again increases and finally increased but below training accuracy until the final epoch.



(a)

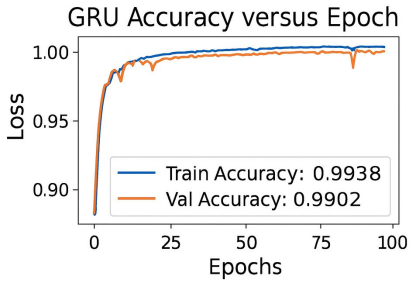


(b)

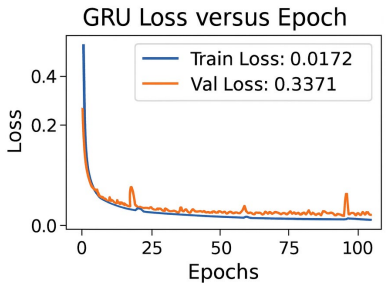
Fig. 27. Accuracy versus loss with their corresponding epochs for multi classification using Bi-LSTM.

When it both reach 99.34% and around 99.12% during the 100th epoch, there is 0.002% gaps between training accuracy and validation accuracy. The start point of training and validation loss curves are shown in Fig. 27 as 0.5 and 0.23, respectively. The validation loss value goes down then experience goes up increase again decrease and training loss value also in the same way.

According to Fig. 28(a) and (b) shows the train and validation progress of the suggested GRU model, the training accuracy value line begins nearly 85% while the validation accuracy value begins at nearly 90%. When it both reach 99.38% and around 99.02% during the 1000th epoch, there is 0.002% gaps between training accuracy and validation accuracy. The start point of training and validation loss curves as 0.5 and 0.23, respectively. The validation loss value goes down then experience goes up increase again decrease and training loss value also in the same way.



(a)



(b)

Fig. 28. Accuracy versus loss with their corresponding epochs for multi classification using GRU.

Extending the above analyses to the multi-class classification scenario, as illustrated in Figs. 29 and 30, the GRU and DNN models demonstrate superior performance, exhibiting higher training accuracy and lower validation loss than the other models, similar to their behavior observed in the binary classification case.

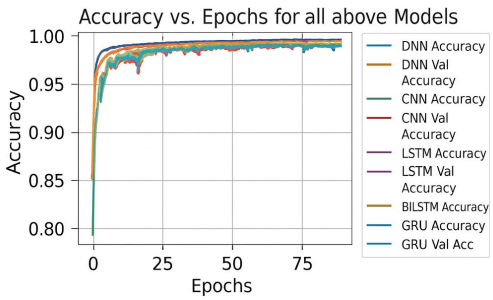


Fig. 29. Accuracy versus epochs for all above models using multi classification.

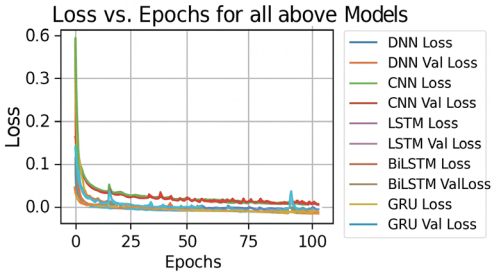


Fig. 30. Loss versus epochs for all above models using multi classification.

G. Test Results

The model is trained on 118,813 and tested on 29,704 for both binary and multi classification. The proposed model evaluated by using accuracy score in our study. As per the performance assessed Bi-LSTM could be the best model for both binary and multi classification of network attack. The following Table VIII, Figs. 31 and 32 which shows the test result of each model.

TABLE VIII. ACCURACY, RECALL, PRECISION AND F1-SCORE FOR BINARY CLASSIFICATION

Models	Accuracy	Recall	Precision	F1-Score
DNN	99.07%	99.16%	99.17%	99.03%
CNN	98.91%	98.53%	98.53%	98.87%
LSTM	98.94%	99%	98.95%	98.89%
Bi-LSTM	99.15%	99.07%	99.07%	99.12%
GRU	99.08%	99.17%	99.18%	99.04%

As shown in Table VIII, among all models used in this study, the Bi-LSTM achieved the best performance in binary classification, obtaining 99.15% Precision, 99.07% Accuracy, 99.07% Recall, and 99.12% F1-Score.

As Fig. 31 also shows, Bi-LSTM is the best model for multi classification when it comes to test accuracy among all the models we utilized for our investigation scoring accuracy of 99.12%.

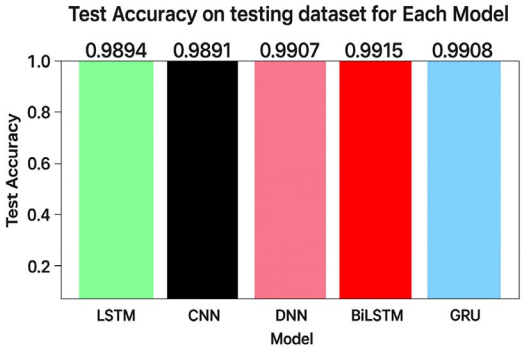


Fig. 31. Test accuracy on testing dataset sample for binary classification.

Table IX shows the model scored best accuracy for both binary and multi classification which is 99.15% and 99.12% respectively. Therefore, this shows that Bi-LSTM is the best performing model from all models used in this study experiments even though all of them outperformed the related works stated in chapter two of this paper.

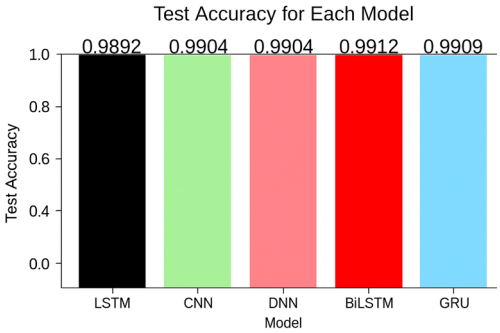


Fig. 32. Test accuracy on testing dataset sample for multi classification.

TABLE IX. TEST ACCURACY FOR BOTH BINARY AND MULTI-CLASSIFICATION

Models	Accuracy	
	For binary classification	For multi classification
DNN	99.07%	99.04%
CNN	98.91%	98.37%
LSTM	98.94%	98.92%
Bi-LSTM	99.15%	99.12%
GRU	99.08%	99.09%

As we have seen under classification reports accuracy value, recall, precision value and F1-Score is 99% for models in binary classification for both classes (normal and attack). However, as the above table shows the test results for all models used in the study, Bi-LSTM outperformed other models.

Even though, in multi classification, recall, precision and F1-score is 1 for class DoS through all models whereas classification report shows accuracy of 99% for models except CNN which is scored 98%. All models in the multi classification report scored recall, precision and F1-Score of 99% for the normal class again except CNN which scored 98% precision and 99% recall as well as F1-Score.

In the multi classification for R2L and U2R most of models scored differently evaluation metrics result. As a result, in order to differentiate the models, we enforced to use test accuracy provided in Table VI. Based on the result, in the table we conclude that Bi-LSTM model is best performing model which is scored test accuracy of 99.12% for multi classification.

V. CONCLUSION

Computer networks have great impacts on public services, economic growth, healthcare, education, social development, and innovation are all significantly impacted by computer networks. Networks are essential to the general progress and well-being of people and communities because they enable efficient communication, information accessibility, and the seamless operation of many sectors.

However, networks are seriously threatened by a number of assaults, such as DoS, Probe, R2L, and U2R, which can result in large-scale data and financial losses. For prompt intervention and control actions to safeguard network security, the ability to identify and diagnose these assaults is crucial.

Four distinct types of network attacks—DoS, Probe, R2L, and U2R—were included in the 148,517 network

records used in this study. The dataset was obtained from a publicly accessible source. The required data preprocessing methods, including cleaning, standardization, train-test split, label-encoding, and one-hot-encoding, are then carried out this research works.

Deep learning methods were adopted for this study. To minimize training time and enhance computational efficiency, the models were trained using a Graphics Processing Unit (GPU) provided by the Google Colab environment.

The primary goal of this study is to construct a model that identify and classify active network attacks using deep learning techniques. Additionally, the study sought to determine which of the five models—DNN, CNN, LSTM, Bi-LSTM, and GRU—performed best in identifying and categorizing current network attacks. In order to accomplish this, we adhered to the particular goals and study questions outlined in this paper.

The study found that Bi-LSTM outperformed the other models, achieving high accuracy to detect and classify active network attacks for binary and multi classification.

The proposed model attained high F1-score, recall value, precision, and accuracy value, making it suitable for practical application in detecting and classifying active network attacks.

Therefore, the study selected Bi-LSTM model as a proposed model. This is because Bi-LSTM has a well-designed architecture and can operate in both forward and backward direction which improves its ability to accurately detect network attack and classify it.

As a contribution, we could experiment with five deep learning models after standardizing the data as needed to see which one can detect and classify active network attacks with a high detection rate or recall of 99.07%, low false positive rates of 0.9% and false negative rates of 0.8%, and high accuracy of 99.15% and 99.12% for binary and multi classification, respectively. Overall, this study demonstrates that the deep learning approach fared better than the earlier research listed under related works in detecting and classifying active network assault.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Writing, Review and Editing, Karthikeyan Kaliyaperumal. Methodology, Raja Sarath Kumar Boddu. Investigation and Validation, Sai Kiran Oruganti. All authors had approved the final version.

References

[1] M. A. T. Padmasiri, V. V. V. Ganepola, R. Herath, L. P. Welagedara, G. Ganepola, and P. Vekneswaran, "Survey on deep learning based network intrusion detection and prevention systems," in *Proc. 13th International Research Conference of General Sir John Kotelawala Defence University,* December 2020.

[2] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–36, 2021. https://doi.org/10.1145/3439950

[3] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," arXiv preprint, arXiv:1901.03407v2, 2019.

[4] R. Kumar, "An overview of computer networking as an introduction OF," *Open Transparent PEER Reviewed & Refereed Research Journal*, July 2023.

[5] J. F. Kurose and W. R. Keith, "Computer networking: A top-down approach edition," *Addision Wesley*, vol. 12, 2007.

[6] W. Yang, "Research on the relationship between computer network and economic development in information environment," *Journal of Physics: Conference Series*, vol. 1744, no. 4, 042011, 2021. https://doi.org/10.1088/1742-6596/1744/4/042011

[7] R. Zeng, B. J. Jansen, H. Liang, and J. Ye, "Development of network security based on information technology," in *Proc. International Conference on Cognitive based Information Processing and Applications (CIPA 2021)*, 2021, pp. 321–328.

[8] Y. Bonaparte, "Global financial stability index," *SSRN*, 2024. https://doi.org/10.2139/ssrn.2753667

[9] C. Ieracitano, A. Adeel, F. C. Morabito, and A. Hussain, "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," *Neurocomputing*, vol. 387, pp. 51–62, 2020. https://doi.org/10.1016/j.neucom.2019.11.016

[10] A. Judith, G. J. W. Kathrine, and S. Silas, "Efficient deep learning-based cyber-attack detection for internet of medical things devices," *Engineering Proceedings*, vol. 59, no. 1, 139, 2023. https://doi.org/10.3390/engproc2023059139

[11] M. Anwer, S. M. Khan, and M. U. Farooq, "Attack detection in IoT using machine learning," *Engineering, Technology and Applied Science Research*, vol. 11, no. 3, pp. 7273–7278, 2021. https://doi.org/10.48084/etasr.4202

[12] S. Aftergood, "Cybersecurity: The cold war online," *Nature*, vol. 547, pp. 30–31, 2017. https://doi.org/10.1038/547030a

[13] Y. Wu, D. Wei, and J. Feng, "Network attacks detection methods based on deep learning techniques: A survey," *Security and Communication Networks*, vol. 2020, no. 1, 8872923, 2020. https://doi.org/10.1155/2020/8872923

[14] M. De Lucia, P. E. Maxwell, N. D. Bastian, A. Swami, B. Jalaian, and N. Leslie, "Machine learning raw network traffic detection," in *Proc. Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III*, 2021, pp. 185–194. https://doi.org/10.1117/12.2586114

[15] V. Malhotra and M. Semwal. (August 2019). Detect malicious & benign websites using machine learning. [Online]. Available: https://www.researchgate.net/profile/Mayank-Semwal-2/publication/334824944_Comparison_of_3_Supervised_Machine Learning_Models/links/5eb2fa7f92851cbf7fad90d4/Comparison-of-3-Supervised-Machine-Learning-Models.pdf

[16] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017. https://doi.org/10.1109/ACCESS.2017.2762418

[17] M. Gao, L. Ma, H. Liu, Z. Zhang, Z. Ning, J. Xu, "Malicious network traffic detection based on deep neural networks and association analysis," *Sensors*, vol. 20, no. 5, 1452, 2020. https://doi.org/10.3390/s20051452

[18] S. Abbas, I. Bouazzi, S. Ojo, A. Al Hejaili, G. A. Sampedro, A. Almadhor, and M. Gregus, "Evaluating deep learning variants for cyber-attacks detection and multi-class classification in IoT networks," *PeerJ Computer Science*, vol. 10, e1793, 2024. https://doi.org/10.7717/peerj-cs.1793

[19] S. L. Ramaswamy and J. Chinnappan, "RecogNet-LSTM+CNN: A hybrid network with attention mechanism for aspect categorization and sentiment classification," *Journal of Intelligent Information Systems*, vol. 58, no. 2, pp. 379–404, 2022. https://doi.org/10.1007/s10844-021-00692-3

[20] M. Kamyab, G. Liu, and M. Adjeisah, "Attention-based CNN and Bi-LSTM model based on TF-IDF and GloVe word embedding for sentiment analysis," *Applied Sciences*, vol. 11, no. 23, 11255, 2021. https://doi.org/10.3390/app112311255

[21] B. Iung, "Coeur et grossesse," *EMC-Traité de Médecine AKOS*, vol. 8, no. 2, pp. 1–4, 2013. https://doi.org/10.1016/s1634-6939(13)59289-1

[22] W. Stallings, *Network Security Essentials: Applications and Standards*, Chennai, India: Pearson Education India, 2011.

[23] D. Hutchison. (2017). Barocchetto. *Oxford Art Online*. [Online]. Available: https://doi.org/10.1093/gao/9781884446054.article.t006431

[24] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O.

Al-Shamma *et al.*, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, 53, 2021. https://doi.org/10.1186/s40537-021-00444-8

[25] B. R. Konatham, "A secure and efficient IIoT anomaly detection approach using a hybrid deep learning technique," M.S. thesis, Department of Computer Science and Engineering, Wright State University, Ohio, USA, 2023.

[26] F. Shahzad, M. Pasha, and A. Ahmad, "A survey of active attacks on wireless sensor networks and their countermeasures," arXiv preprint, arXiv:1702.07136, 2017.

[27] A. A. Salih, S. Y. Ameen, S. R. M. Zeebaree, M. A. M. Sadeeq, S. F. Kak, N. Omar *et al.*, "Deep learning approaches for intrusion detection," *Asian Journal of Research in Computer Science*, vol. 9, no. 4, pp. 50–64, 2021. https://doi.org/10.9734/ajrcos/2021/v9i430229.

[28] A. A. Waoo and B. K. Soni, "Performance analysis of sigmoid and relu activation functions in deep neural network," in *Proc. SCIS 2021*, Singapore, 2021, pp. 39–52. https://doi.org/10.1007/978-981-16-2248-9_5

[29] M. T. Tun, D. E. Nyaung, and M. P. Phyu, "Network anomaly detection using threshold-based sparse autoencoder," in *Proc. 11th International Conference on Advances in Information Technology*, 2020, pp. 1–8. https://doi.org/10.1145/3406601.3406626

[30] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*, 2009, pp. 1–6. https://doi.org/10.1109/CISDA.2009.5356528

[31] J. Terven, D. M. Cordova-Esparza, J. A. Romero-González *et al.*, "A comprehensive survey of loss functions and metrics in deep learning," *Artif. Intell. Rev.*, vol. 58, no. 195, 2025. https://doi.org/10.1007/s10462-025-11198-7

[32] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 266–282, 2014. https://doi.org/10.1109/SURV.2013.050113.00191

[33] A. Chatterjee and B. S. Ahmed, "IoT anomaly detection methods and applications: A survey," *Internet of Things*, vol. 19, 100568, 2022. https://doi.org/10.1016/j.iot.2022.100568

[34] C. M. Hsu, H. Y. Hsieh, S. W. Prakosa, M. Z. Azhari, and J. S. Leu, "Using long-short-term memory based convolutional neural networks for network intrusion detection," in *Proc. the 2018 International Wireless Internet Conference*, 2019. https://doi.org/10.1007/978-3-030-06158-6_9

[35] O. A. Sarumi, A. O. Adetunmbi, and F. A. Adetoye, "Discovering computer networks intrusion using data analytics and machine intelligence," *Scientific African*, vol. 9, e00500, 2020.

[36] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, 2018, pp. 108–116. https://doi.org/10.5220/0006639801080116

[37] I. Ahmad, M. Imran, A. Qayyum, M. S. Ramzan, and M. O. Alassafi, "An optimized Hybrid Deep Intrusion Detection Model (HD-IDM) for enhancing network security," *Mathematics*, vol. 11, no. 21, 4501, 2023. https://doi.org/10.3390/math11214501

[38] A. D. Aguru and S. B. Erukala, "A lightweight multi-vector DDoS detection framework for IoT-enabled mobile health informatics systems using deep learning," *Inf. Sci.*, vol. 662, 120209, 2024.

[39] C. Igwenagu, "Fundamentals of research methodology and data collection," *Research Methodology*, vol. 4, no. 11, pp. 4–9, 2016

[40] F. S. Alrayes, M. Zakariah, M. Driss, and W. Boulila, "Deep Neural Decision Forest (DNDF): A novel approach for enhancing intrusion detection systems in network traffic analysis," *Sensors*, vol. 23, no. 20, 8362, 2023. https://doi.org/10.3390/s23208362

[41] A. Aldhaheri, F. Alwahedi, M. A. Ferrag, and A. Battah, "Deep learning for cyber threat detection in IoT networks: A review," *Internet Things Cyber-Phys. Syst.*, vol. 4, pp. 110–128, 2024.

[42] T. Al-shehari and R. A. Alsowail, "An insider data leakage detection using one-hot encoding, synthetic minority oversampling and machine learning techniques," *Entropy*, vol. 23, no. 10, 1258, 2021. https://doi.org/10.3390/e23101258