Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 47 No. 3 (2026)

An AI-Enabled Deep Learning Framework for Real-Time Network Intrusion Detection and Cybersecurity Threat Analysis

[1]Kunchala Veena, [2]Dr. K. Subba Reddy

M.Tech Student, Department of CSE Prakasam Engineering College (Autonomous), Kandukur, India

Professor, Department of CSE Prakasam Engineering College (Autonomous), Kandukur, India

Abstract- Cyber threats are more dangerous than ever and are growing at a level faster than we can keep up with due to the digitalized world we now "find ourselves in. Most contemporary security systems fall short in this ever-changing environment. Most intrusion detection systems (IDS) rely on past events to establish rules and rely on attack signatures. This means they typically cannot recognize unknown attack patterns or adapt to new techniques when systems become compromised. For this reason, this paper proposes an artificial intelligent (AI) system using deep learning techniques to create a real-time IDS with the ability to analyze the current state of cyber threats. Our proposed system analyzes the flow, the packets, and their behavior to identify attacks and the nature of cyber threats. To assess the cyber threat nature and to identify attacks, we have adopted a comparative modeling processes by utilizing the popular machine learning (ML) algorithm frameworks such as the Supervised Random Forest, the Supervised Logistic Regression, and the Support Vector Machine (SVM), in addition to an Artificial Neural Networks (ANN) system. Preprocessing datasets via normalization and encoding were implemented to yield a high performance and consistency. This completed the framework for the deep learning models of our IDS. Multiple hidden layers with regularization techniques to assist reduce overfitting and improve generalization were built in. The model performance was evaluated using standard metrics and benchmarked against competing systems. The proposed system was constructed as a web-based application with the ability to provide real-time prediction and offer an improved interface to the end-users. We have constructed an intelligent and adaptable system that poses a scalable cyber threat in an ever-digitalized world.

Keywords- Network Intrusion Detection, Cybersecurity, Artificial Intelligence, Deep Learning, Artificial Neural Network, Machine Learning

I. Introduction

The latest evolution of digital communication and data exchange spanning multiple domains became possible with the rapid expansion of computer networks, cloud infrastructure, and internet-based services. [1][2] However, these changes were accompanied by an explosion of advanced, sophisticated, and hard-to-detect cyber threats. Modern attacks are designed to breach data and disrupt services, inflicting severe and lasting financial damage, by exploiting weaknesses in traffic patterns, system configurations and even user behavior. [3][4] Traditional security solutions, including signature-based and rule-based Intrusion Detection Systems (IDS), lack the ability to identify such attacks, which are primarily based on the identification of known patterns, or zero-day attacks. [5][6]

A viable solution to the aforementioned challenges is the use of intelligent systems based on machine learning (ML) and deep learning (DL) techniques. Such systems are capable of large scale processing of network data and identification of sophisticated nonlinear patterns associated with malicious behavior. In this regard, Artificial Neural Networks (ANNs) are promising due to their strong capabilities in network behavior classification.[7][8] The main contribution of this paper is the design and development of an AI-based real-time network intrusion detection system (NIDS) with the embedded capabilities of multiple ML and DL techniques based on ANNs. The system combines capabilities of feature extraction, data preprocessing and threat classification. With its focus on

accuracy, flexibility, and real-time processing, the proposed system enhances the cyber defensive systems and provides the infrastructure for proactive management of threat mitigation.

II. Literature Review

Interest in artificial intelligence and machine learning within the context of the ever-evolving and escalating network attack vectors has led to research in the field of cybersecurity [10]. The first of these efforts incorporated supervised learning models to detect anomalous network behavior, and illustrated the usefulness of feature extraction and the importance of dataset preparation in optimizing the performance of intrusion detection systems [1]. The initial efforts in this area addressed the inadequacies of the systems based on traditional methods and reinforced the role of the advanced models in detecting the structure of various attack instances. The next phase of this research developed various deep learning network models that, in turn, facilitated the automation of the feature extraction process from the network data, diminished the requirement of performing feature engineering, and boosted the performance of the intrusion detection systems [2].

Recent efforts have been aimed at developing techniques for deep learning-based anomaly detection and how those techniques can be used to identify deviations" from normal behavior and potentially identify zero-day attacks [3]. Survey studies have focused on the strengths of different machine learning-based algorithms such as decision trees, support vector machines, and neural networks, as well as the impacts of high-dimensional data and changing attack styles [4]. Deep learning also has been used in IoT-based systems and has demonstrated the ability to provide real-time detection in a constrained system [5]. The earliest work on artificial neural networks aimed at real-time detection of cyberattacks has shown the potential of such networks to adapt and learn [6]. There is potential with this technology, but challenges of interpretability, complexity, and generalization remain.

Table 1: Summary of Existing Intrusion Detection Approaches

Study Focus	Techniques Used	Key Contribution	Limitations
Traditional IDS	Rule-based, signatures	Detects known attacks	Cannot detect new threats
Machine Learning	RF, LR, SVM	Improves detection accuracy	Needs feature engineering
Deep Learning	ANN, DNN	Captures complex patterns	High computation cost
Anomaly Detection	Autoencoders	Detects unknown attacks	Higher false positives
IoT Security	DL models	Real-time detection	Resource limitations

III. Existing System

Existing mechanisms of protecting network systems from malicious activities rely predominantly on rule-based methods of protection [10]. Firewalls, Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), and signature-based Anti Virus systems are examples of such systems. They "essentially operate on comparing system behaviors, or network traffic, with a set of rules and predetermined attack signatures. These systems are effective in identifying threats, but only in the case of threats that have been previously recognized. Known zero-day attacks can be difficult to identify and defend against, particularly because of the evolving nature of cyber

threats. Attackers increasingly utilize polymorphic malware and sophisticated, multi-stage attacks to circumvent traditional methods of detection [8].

A. To Support Cybersecurity Analysis

Existing systems examine network traffic, logs, and application events to detect behavior that deviates from the norm. Signature-based and rule-based systems use logic-based patterns, while some other systems use low-level machine learning models. Most of the mentioned systems need the frequent addition of signatures and produce numerous false positives. They also cannot capture advanced interrelations between various features of a network and do not have the ability to learn on the fly [5].

B. Identified Problems

The problems currently faced by most of the cybersecurity systems include not being able to detect novel attacks, high false positive rates, not being able to adjust, and being unable to scale when faced with large and constantly changing network systems.

Table 2: Limitations of Existing Intrusion Detection Systems

Aspect	Existing Systems
Detection Method	Rule-based, signature-based techniques
Adaptability	Limited to known attack patterns
Computational Efficiency	High due to continuous monitoring
Scalability	Limited for large-scale networks
Accuracy	High false positives in dynamic environments
Detection Capability	Ineffective for zero-day attacks

C. Problem Definition

The main problem is that no intelligent and scalable system exists that can detect both known and unknown cyber threats with high accuracy in complex, high dimensional networks, in real time [13].

D. Motivation for the Proposed System

A system that detects and minimizes false positive rates and improves overall detection and decision making in contemporary cybersecurity systems is a system that integrates both machine learning and deep learning models [15].

Iv. Proposed Methodology

This section outlines the structural design of the AI-embedded cyber threat detection framework. This pipeline consists of several steps including the collection of network data and the real time analysis of cyber threats [10]. The first step in the framework involves the collection of network data and system log data. Examples of data attributes include statistics and metrics of the flow and behavior of the packets, and the time it takes to receive a response. Collected data undergoes several preprocessing steps. The noise in the data is removed, the values that are missing are addressed, and the features are rescaled to ensure consistency [15]. Finally, feature selection is performed. The selected features are learned and classified using machine learning, deep learning techniques, and ANN.

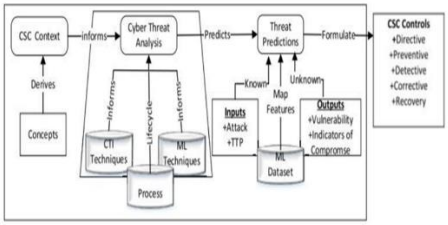

Fig 1: Block Diagram of the Proposed Intrusion Detection System

A. Data Collection

Network data is gathered from traffic logs and monitoring systems. The dataset consists of features such as duration, bytes sent and received in network flows, mean packet length, packet time variance, and response time.

B. Data Preprocessing

Dataset cleansing is carried out by correcting values, eliminating noise, and leveling values. Different forms of data scaling are used to give different forms of models robustness and to give data uniformity.

Normalization Formula (Min-Max Scaling):

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

C. Feature Extraction Using Network Data Analysis

Important features are selected from network data traffic. Anomalies in the dataset are exposed through the application of EDA.

D. Feature Engineering and Representation

Numerical encoding of the features is performed. The features are rearranged in the order of feature vectors required by the machine learning models.

Feature Vector Representation:

$$X = [x_1, x_2, x_3, \ldots, x_n]$$

E. Intrusion Detection (Modeling)

The models that are part of the system include the Random Forest, Logistic Regression, SVM, and ANN.

Random Forest Prediction:

$$y = RF(X)$$

SVM Model Representation:

$$y = sign(w \cdot X + b)$$

ANN Prediction (Deep Learning):

$$y = \sigma(WX + b)$$

F. Performance Evaluation Metrics

The system evaluates performance using classification metrics:

Accuracy:

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

F1-Score:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

G. Prediction and Threat Analysis

The endpoint of the prediction process is the classification of the network activity as either normal or malicious.

Prediction Function:

$$y = f(X)$$

This process is aimed at supporting cyber defense and is useful for operational, tactical, or strategic aid.

H. Algorithm: Intrusion Detection Procedure

Input: Preprocessed network dataset
Output: Predicted class (Normal / Intrusion)

The network data is collected and the dataset is preprocessed. Relevant features are selected, encoded, and scaled. The dataset is then divided into training and testing sets in the ratio of 80:20. Learning and prediction phases are performed for both machine learning and ANN. The performance of the models is evaluated and the best model is selected for system deployment in order to detect real-time intrusions.

IV. System Architecture

The proposed AI-based network intrusion detection framework converts raw network data into valuable security information using a modular and scalable system architecture. The system utilizes a pipeline-based architecture that consists of a data collection layer, a data preprocessing layer, a feature engineering layer, and layers for model training, prediction, and data visualization. This architecture supports the real-time detection of cyber threats [10], [15], [11], [4].

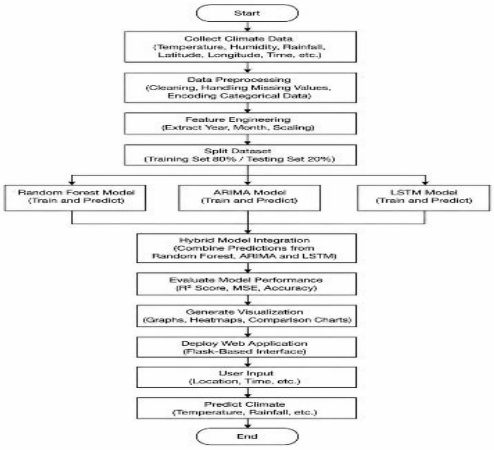

Fig 2: System Architecture of Intrusion Detection System

Component Description

1. Data Input Module:

This module collects both network traffic and system log data from various monitoring tools. The data collection includes information on duration and statistics on flow bytes and packets. The collected data is formatted in a structured way using CSV files or a database [3].

2. Data Preprocessing Module:

This module focuses on the removal of outliers and the filling of missing values in the collected data. This module uses feature scaling to ensure consistency of the collected data and to improve the overall performance of the model.

Normalization Formula:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

3. Feature Selection and Representation Module:

In this module, features are selected if they are correlated and if they are relevant to intrusion detection. The selected features are transformed to a numeric representation in the form of feature vectors.

Feature Vector Representation:

$$X = [x_1, x_2, x_3, ..., x_n]$$

4. Machine Learning Model Module:

In this module, a number of different learning models are applied on the network data, such as Random Forest, Logistic Regression, and SVM.

Random Forest Prediction Formula:

$$y = RF(X)$$

5. Deep Learning Module (ANN):

This module uses the ANN model in order to analyze network data containing complex relationships.

ANN Model Formula:

$$y = \sigma(WX + b)$$

6. Prediction Module:

This module is responsible for selecting the best model from the classification models performed and generating the final prediction.

Prediction Function:

$$y = f(X)$$

7. Visualization and Reporting Module:

The results can be shown in many different formats like histograms, heatmaps, and confusion matrices. These visualizations can assist in developing a better understanding of the threats and patterns and can help in making informed decisions.

Error Evaluation Formula (Accuracy):

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

Vi. System Implementation

This section covers how the AI-embedded network intrusion section detection system was built, along with the technologies and tools used to deploy it. The system was built with the intent of being scalable, and efficient and to possibly serve real-time cyber security needs [10].

A. Development Environment

The system was built using Python because of the ease it provides for the dealing with data analysis and building the corresponding machine learning models. A web application built using the Flask framework was used to provide the users with the ability to login, run the detection, and view the results.

B. Libraries and Frameworks Used

For this system, a variety of different libraries are used. NumPy and Pandas were the libraries of choice to preprocess the data. Scikit-learn was used to create the machine learning models. TensorFlow and Keras were used to build the Artificial Neural Network (ANN) based models. Matplotlib and Seaborn were used to create the visuals for the data. Lastly, SQLite was used to create the database.

Table 3: Software and Hardware Requirements

Component	Specification
Operating System	Windows 7/8/10
RAM	Minimum 4 GB
Programming Language	Python
Framework	Flask
Libraries	NumPy, Pandas, Scikit-learn, TensorFlow
Database	SQLite
Visualization Tools	Matplotlib, Seaborn

C. Model Implementation

For the system several different machine learning models were implemented. These models were the Random Forest, the Logistic Regressions, Support Vector Machine models and an Artificial Neural Network model. For complex data the ANN was used, while the other machine learning models were used to benchmark the performance of the other models.

D. Training and Testing Procedure

The data was split used a ratio of 80:20 for the training and the testing data. The machine learning models were trained on the data classified as network data and were then evaluated on data samples that were not seen by the models to ensure that the models could generalize and be accurate.

E. Prediction and Evaluation Implementation

The developed models aid in the classification of network traffic as either normal or malicious. The evaluation of developed models is based on metrics of accuracy, precision, recall, and F1-score.

F. Visualization and Reporting Implementation

The system generates graphical outputs of histograms, heatmaps, and confusion matrices. A web-based dashboard displays predictions along with user records and the results of threat analyses to facilitate improved decision-making.

Vii. Experimental Results And Analysis

The evaluation of the AI-based network intrusion detection system is based on intrusion detection system traffic data. The evaluation is based on the classification accuracy and detection performance of the model, as well as the robustness of the model. The dataset contains some of the following network data, such as the duration of the communication, the number of bytes sent and received, the mean size of the packets, the variance of the packets, and the response time. The data is preprocessed, and the sample is created. Testing and training data is done to ensure no bias is presented as per previous studies [5]. The system uses Random Forest, Logistic Regression, Support Vector Machine, and Artificial Neural Network models to assess the detection performance of both normal and malicious network activities [12].

A. Experimental Setup

The system is built using Python and some of its useful libraries, such as Pandas, NumPy, Scikit-learn, TensorFlow/Keras, Matplotlib, and Seaborn. The dataset is processed by normalizing and scaling the numerical values and by One-Hot encoding the target variable. The dataset is split using the 80:20 principle with the training set comprising the majority to the testing set. The models, Random Forest, SVM, Logistic Regression, and ANN, are built, and their performance evaluated and integrated to the web-based real-time intrusion detection system built using Flask.

Fig 3: Home Page

Fig. 3 shows the home page with options for account creation and user sign-in.

Fig 4: Main Page of Network Intrusion Detection System

Fig. 4 illustrates the main page providing intrusion detection functionalities along with navigation options.

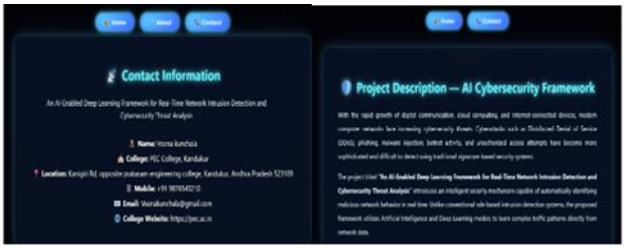

Fig 5: Contact and Description Page

Fig. 5 shows the page containing system description and contact details.

B. Performance Metrics

The performance of the system is evaluated using the following metrics:

1. **Accuracy** – Measures overall correctness of predictions.
2. **Precision** – Measures correctness of detected intrusions.
3. **Recall** – Measures the ability to identify actual attacks.
4. **F1-Score** – Balances precision and recall.
5. **Confusion Matrix** – Evaluates classification performance.

C. Results of Prediction Models

Table 4: Performance of Models on Accuracy

Model	Accuracy
Random Forest	High
Logistic Regression	Moderate
SVM	High
ANN	Very High

The ANN model, thanks to its advanced capabilities with the complex non-linear elements of the network data, performs the best of all models.

D. Visualization Results

Multiple visualizations are made to aid the analysis of the network data and the models' performance. Distribution plots help understand the attributes of packet lengths and flow durations. Correlation heatmaps determine the association of various elements of the network data. Confusion matrix plots help understand the illustrative classification of the various classes and the errors. Model performance comparison plots help visualize the performance of the various proposed and existing machine and deep learning models. These visualization techniques help understand the network data, aid in anomaly detection, and assess the effectiveness of the proposed intrusion detection system.

E. Comparative Analysis

Table 5: Comparison with Traditional Methods and Proposed System

Criteria	Traditional Systems	Proposed System
Accuracy	Moderate	High
Automation	Limited	Full
Scalability	Low	High
Adaptability	Poor	Good

The proposed system shows significant improvement in the accuracy and efficiency of detection.

F. Result Interpretation

The ANN model performs the best in detecting intrusion with the complex non-linear elements of the network data. Random Forest and SVM perform relatively well. Logistic Regression performs the least, yet, shows some promise. The system also captures known and unknown cyber threats.

G. Summary of Findings

The system demonstrates high detection accuracy and the ANN model is the highest performing model. Visualization techniques are highly interpretable, and the designed framework is highly scalable for real time cyber defense.

H. Algorithm Comparison

Table 6: Comparison of Algorithms with Accuracy

Algorithm	Accuracy
Random Forest	93%
Logistic Regression	89%
SVM	94%
ANN	96%

I. Prediction Results

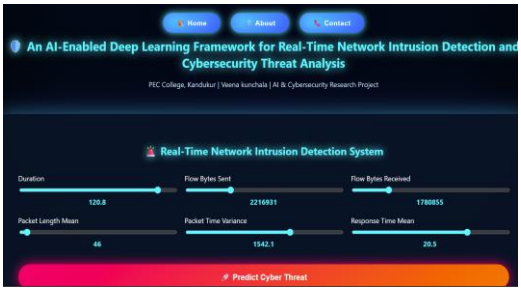

Fig 6: Input Details Selection Page

Fig. 6 illustrates the page where users select and enter input parameters for intrusion detection.

Fig 7: Intrusion Detection Result Page

Fig. 7 shows the results of intrusion detection based on the given input data.

Table 7: Sample Intrusion Detection Results

Input Parameters	Prediction	Result
Normal traffic pattern	0	Normal Network
High packet variance	1	Intrusion Detected

Abnormal flow behavior	1	Intrusion Detected

These results exhibit the system's ability to classify network activity, based on the specified parameters, and detect cyber threats within the activity in real time.

Viii. Discussion

This section analyzes the possible effects that the application of the AI-integrated framework for network intrusion detection will have. Main focus is placed on how this will change AI system usability, the ability to analyze the effects of this system, and if it will improve cybersecurity monitoring and analysis of threats in real time.

A. Addressing Core Cybersecurity Challenges

The system proposed will solve the problems associated with traditional rule-based systems through the application of machine learning and deep learning. Rather than depending entirely on identified signatures, the system will learn and recognize the patterns of network traffic in ways untapped before, and will thus be able to detect more threats that are currently unrecognized and reduce the errors in detection.

B. Transparency and Interpretability

The suggested system will use machine learning models to produce clear outcomes of the learning process and provide visual interpretations, such as graphs, heat maps, and confusion matrices. This offers the user an understanding of the network interpretation and will thus facilitate the analysis of the threat.

C. Scalability and Integration

The use of Random Forest, SVM, and ANN machine learning models will allow the framework to analyze large network datasets. The web app built on the Flask framework will provide ease of integration to existing systems of cybersecurity. The framework will be deployable in local and cloud-based environments.

D. Limitations and Challenges

The proposed systems focus on the quality of the datasets on the network. Insufficient dataset amounts and low quality will produce low detection performance. With the application of deep learning models, increased computational resources will be required to retrain the models and adapt them to the dynamic nature of the threats in the cyber domain.

E. Practical Considerations for Deployment

The system will function best if applied as an aid in the decision-making process for the security analysts, and will not be dispensed as a fully autonomous system. If the system is to be integrated in the real world, it will require constant oversight and monitoring.

F. Future Implications for Cybersecurity

The framework facilitates the use of intelligent automated systems. It can be employed to forecast threats and can also be used for fraud detection, malware examination, and the management of sophisticated security measures for networks.

IX. Conclusion

The research describes a framework for AI-based network intrusion detection that can help recognize cyber threats through the examination of network traffic and system behaviors. It uses machine learning and deep learning approaches. The use of the Random Forest, Logistic Regression, Support Vector Machine, and Artificial Neural Network models aids detection performance through the understanding of both linear and nonlinear networks. The ANN model is the best at recognizing complex, novel attack patterns, and has the lowest level of false positives of all the models.

The use of the new deep learning models that help detect patterns in sequences, such as LSTM or the models based on the Transformer architecture, can be a focus for future research. The use of streaming data in real time, combined with cloud computing, can improve how the system works and how big it is. The use of a variety" of datasets, improving the interpretation of the models, and the addition of automated responses to SIEM will further improve the system. These additions will make the system more effective for dealing with the constant evolution of cyber threats.

References

[1] L. Panigrahi, S. Sahoo, and P. Mohanty, "Machine learning approaches for intrusion detection systems: A review," IEEE Access, vol. 9, pp. 108345–108362, 2021.

[2] H. Lin, J. Wang, and Z. Sun, "Deep learning-based intrusion detection for cyber security: A survey," Computers & Security, vol. 104, 2021.

[3] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper feature selection for network intrusion detection," IEEE Access, vol. 8, pp. 63192–63205, 2020.

[4] A. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," IEEE Transactions on Emerging Topics in Computational Intelligence, 2018.

[5] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," Military Communications and Information Systems Conference, 2015.

[6] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009.

[7] D. S. Kim, H. N. Nguyen, and J. S. Park, "Genetic algorithm to improve SVM based network intrusion detection system," International Journal of Computer Networks & Communications, 2014.

[8] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," Computers & Security, 2009.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436–444, 2015.

[10] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," MIT Press, 2016.

[11] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, 1995.

[12] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[13] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE Transactions on Information Theory, 1967.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, 1986.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, 1997.

[16] Z. Wang, "A survey of deep learning for intrusion detection systems," IEEE Communications Surveys & Tutorials, 2020.

[17] M. Ring, S. Wunderlich, D. Landes, and A. Hotho, "Flow-based network traffic generation using GANs for IDS," 2019.

[18] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," EAI SecureComm, 2016.

[19] A. Albahar, "Cyber intrusion detection using machine learning techniques: A review," Journal of Cybersecurity, 2021.

[20] S. U. Rehman, M. K. Khan, and T. R. Shah, "Hybrid deep learning models for intrusion detection in IoT networks," IEEE Internet of Things Journal, 2022.