International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Hybrid Intrusion Detection System (IDS) Using Machine Learning and Deep Learning

Aryan Yogi[1], Dr. Jyoti[2]

[1]*Department of Artificial Intelligence and Data Science, University School of Automation and Robotics, Delhi, India*

[2] *Department of Artificial Intelligence and Data Science, University School of Automation and Robotics, Delhi, India*

EMAIL: [1]*yogiaryan20@gmail.com* [2]*Jyoti.usar@ipu.ac.in*

Abstract - This work offers a Hybrid Intrusion Detection System (HIDS) that combines traditional machine learning and deep learning methods for efficient and scalable network attack identification. The system makes use of Principal Component Analysis (PCA) for reducing dimensionality and then utilizes a hybrid CNN-LSTM architecture for feature learning as well as classification. An ensemble method is also utilized to combine Random Forest with the CNN-LSTM to add robustness as well as generalization. The CICIDS2018 dataset, comprising modern real-world network traffic situations, is employed for testing. Our system detects with an accuracy of 99.1% on the test set, far better than conventional classifiers. This paper proves the efficacy of integrating statistical feature engineering with deep sequential models and ensemble techniques to counter cybersecurity attacks in real-time settings.

Key Words*: *Intrusion Detection System, Machine Learning, Deep Learning, CNN-LSTM, PCA, Ensemble Learning, CICIDS2018

1.INTRODUCTION

With the rapid expansion of digital services, cloud computing, and IoT enabled devices, networks are becoming prime targets for cybercriminals. The growing complexity and diversity of modern-day cyberattacks, including Advanced Persistent Threats (APTs), Zero-Day Exploits, and Distributed Denial of Service (DDoS), present a significant challenge to conventional security systems. Traditional Intrusion Detection Systems (IDS), being usually based on static signature-based methods or naive anomaly detection algorithms, are incapable of handling evolving threats. Their lack of adaptability and reliance on pre-defined rules make them vulnerable to emerging attack techniques. This paper proposes the development of a next-generation, hybrid Intrusion Detection System that best leverages the strength of both ML and DL paradigms. With the integration of ML-based feature engineering and preprocessing with DL-based traffic analysis and classification, and boosting predictions through ensemble learning, the system promises to significantly enhance detection accuracy, reduce false positives, and deliver effective high-dimensional network data processing.

The primary objectives of this work are aimed at developing an optimal hybrid Intrusion Detection System (IDS). First, the network traffic data is preprocessed and normalized through Principal Component Analysis (PCA) to facilitate learning efficiently by reducing dimensionality. Subsequently, a CNN-LSTM model is trained and constructed to learn both the spatial and temporal characteristics typical in network traffic data. For even greater increase in robustness, this deep learning model is also combined with a Random Forest classifier in an ensemble strategy. Finally, the performance of the given hybrid IDS is thoroughly tested on the CICIDS2018 dataset, with exhaustive metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.

2.LITERATURE REVIEW

This part presents a systematic overview of recent research on the topic of Intrusion Detection Systems (IDS) using machine learning and deep learning-based methodologies. It gives a description of essential methodologies, benchmark datasets, and state-of-the-art methodologies that have had a significant impact on the formulation of the suggested hybrid IDS model. A perusal of literature demonstrates several methods for feature extraction, including PCA and statistical feature engineering, that have been successful in simplifying network traffic data. Machine learning models such as Random Forest, Support Vector Machines, and deep learning architectures like CNNs and LSTM networks have also been utilized with success in traffic classification and anomaly detection. This review also identifies the advantages of combining dimensionality reduction methods with deep learning models, as well as the advantageous effects of ensemble learning techniques on enhancing model generalization and resilience. These findings provided the groundwork for the hybrid model proposed that integrates PCA, CNN-LSTM, and Random Forest to boost intrusion detection accuracy. The work and developments relevant to this are listed in Table 1.

Paper	Features	Methods	Findings	Limitations
Ingre & Yadav [1]	KDDCup99	Random Forest	A basic Random Forest model obtained solid baseline results on traditional intrusion datasets.	Does not perform well on contemporary, encrypted, or more sophisticated traffic patterns.
Aljawarneh et al. [2]	NSL-KDD	SVM + Naive Baye	Was able to achieve 99.71% accuracy using ensemble voting, showing the robustness of a	NSL-KDD is old, and the results may not generalize well to contemporary or realistic traffic.

		s + KNN	ggregating the out puts of various simple classifiers.	
Meidan et al. [3]	IoT traffic	Deep Autoencoders	Unsupervised deep autoencoders identified IoT-specific attacks efficiently, even without labeling.	Proprietary dataset was not publicly available, reducing reproducibility and benchmarking.
Dir o & Chilamkurti [4]	CICIDS 2017	Distributed DL (Fog)	Distributed deep learning at the network edge improved detection velocity and decreased central processing load.	Restricted resources on edge nodes limit scalability and model sophistication.
Lopez-Martin et al. [5]	KDD99	LSTM	LSTM performed well in modeling sequential behavior in network traffic, applicable to anomaly detection.	Poor performance in multiclass environments and old dataset limited practical applicability.
Sultana et al. [6]	CICIDS 2017	RF, GB, AdaBoost (Ensemble)	Ensemble learning decreased false positives and enhanced precision in identifying known attack patterns.	Didn't test performance on unknown or zero-day types of attacks.
Vinayakumar et al. [7]	UNSW-NB15	CNN, RNN, LSTM	LSTM worked better in learning long-term dependencies and provided greater recall rates for	Deep sequential models led to extremely long training times and large memory usage.
			intrusion detection.	
Wu et al. [8]	NSL-KDD	CNN-GRU	The combination of CNN for spatial features and GRU for sequential data improved multiclass classification performance.	High computational expense because of GRU layers; might not be appropriate for real-time applications.
Shahid et al. [9]	NSL-KDD	Stacked LSTM	Stacked LSTM networks effectively modeled sequential patterns in traffic data, improving detection rates significantly.	Training is hindered by deep architecture and vanishing gradient risk.
Mohammadi et al. [10]	CICIDS 2017	Autoencoder + RF	Employing autoencoders for dimensionality reduction prior to classification enhanced detection rates and eliminated feature noise.	Lack of optimization within the autoencoder's architecture resulted in suboptimal performance.
Kim et al. [11]	NSL-KDD	CNN + Attention	Attention mechanism enhanced attention on relevant features in imbalanced datasets, improving detection accuracy.	Interpretability of models is affected; difficult to justify decisions to stakeholders or system admins.
Ahmed et al. [12]	CICIDS 2018	RF + LSTM + Feature Ranking	Feature selection enhanced model efficiency and enhanced both F1-score and accuracy for attack detection.	Model was not validated in real-world scenarios, with practical deployment concerns.

Roy et al. [13]	BoT-IoT	CNN-LSTM	Temporal and spatial feature extraction achieved high accuracy in identifying DoS and DDoS attacks.	Identification of minority class attacks (e.g., reconnaissance) was poor owing to class imbalance.
Md. Jahid Hasan et al. [14]	CIC IDS 2018	PCA + RF, XGBoost, ANN	Hybrid models achieved enhanced accuracy and F1-score compared to single models by harnessing both feature reduction and ensemble learning.	Manual feature engineering and tuning decrease scalability and restrict end-to-end automation.
Shapoor Zarrin et al. [15]	CIC IDS 2017	BiLSTM, CNN-BiLSTM	CNN-BiLSTM attained 99.96%accuracy, demonstrating high potential for identifying intricate intrusion patterns in time-series data.	Deep models such as CNN-BiLSTM can overfit if not trained on enough diverse data; needs proper regularization.

3.METHODOLOGY

This paper presents a machine learning and deep learning-based hybrid Intrusion Detection System (IDS) to correctly identify cyberattacks in network traffic. The approach is a structured pipeline of data preprocessing, feature extraction, deep learning-based classification, ensemble learning, and performance evaluation. The proposed IDS architecture was implemented and tested using the CICIDS2018, a widely used benchmark dataset for intrusion detection research.

The CICIDS2018 dataset, offered by the Canadian Institute for Cybersecurity (CIC), mimics actual enterprise network traffic and covers benign activity and varied attack types such as DDoS, brute-force, infiltration, botnet, and web attacks. The dataset contains more than 80 features, including flow duration, packet statistics, protocol flags, and header information. It is organized in CSV format, with each row representing a network flow and each column representing extracted features or a class label.

The data is loaded and preprocessed once the libraries have been imported. Preprocessing involves cleaning the raw data, standardizing it, and encoding it to prepare it for machine learning algorithms. Missing values are handled with imputation methods like mean substitution, and redundant or non-informative features—such as timestamps and string-based fields—are dropped. Categorical fields like protocol types are encoded into numerical form using label encoding. After being cleaned, the data is then normalized with z-score standardization to ensure all the values of the features are on an even scale to facilitate better convergence rate and accuracy in deep learning models.

After preprocessing, dimensionality reduction is achieved through Principal Component Analysis (PCA), a proven statistical method that maps the initial high-dimensional feature space to a lower-dimensional one by determining principal components that capture the highest variance in the data. PCA serves to minimize computational complexity, remove noise, and enhance generalization by keeping only the most informative components, thereby preventing overfitting in later learning phases. In this research, the number of components was chosen to preserve about 95% of the data variance to ensure that significant patterns useful in network intrusion detection are maintained.

Principal Component Analysis or PCA is another statistical technique performed to reduce the dimensionality: It transforms an original high dimensional feature space to a lower dimension one by capturing the principal component explaining the highest variance in data. PCA contributes to reducing noise, computational complexities, and thereby improves generalization while retaining more informative components over subsequent learning, which reduces overfitting chances. In this research, the number of features was chosen to keep about 95% of data variance in order to maintain significant patterns important to network intrusion detection.

After reducing the feature space, a hybrid deep model is trained through the interaction of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) layers. CNN is used initially to identify spatial patterns and local correlations among the input features, which can expose the structural signatures of particular attack types. The output of the CNN layers is subsequently fed into LSTM layers, which are particularly effective at learning sequential dependencies and temporal behaviors in the data, such as multi-stage attack sequences or slow-moving infiltration attempts. This deep learning design is formulated to capture both spatial patterns and temporal trends of network traffic, providing an end-to-end view of traffic behavior.

Concurrently with the deep learning pipeline, a Random Forest classifier is also trained over the same preprocessed and PCA-reduced feature set. Random Forest, which is one of the most used ensemble machine learning methods, learns multiple decision trees at training and combines their results to achieve prediction accuracy and insensitivity improvement. Because Random Forest can handle both structured and table data effectively as well as maintain resistance against overfitting, it can provide a powerful complementing classifier to the hybrid method.

The last step of the methodology is to merge the output of the CNN+LSTM model and the Random Forest classifier through an ensemble scheme. The ensemble process utilizes majority voting, where the predicted class is made based on agreement between both models. When both classifiers are in agreement about the prediction, the choice is direct; when there is a disagreement, the majority class among ensemble predictions is utilized. This combination approach minimizes false positives and improves the overall detection rate by taking advantage of the strengths of both deep and conventional learning approaches.

To assess the performance of the proposed hybrid IDS, a number of classification metrics are calculated, such as accuracy, precision, recall, and F1-score. Accuracy calculates the overall prediction correctness, precision calculates the positive classification correctness, recall calculates the model's detection of real attacks, and F1-score calculates the balance between precision and recall, especially for dealing with imbalanced data. Besides numerical values, graphical aids like confusion matrices and Receiver Operating Characteristic (ROC) curves are employed to offer insights into model performance against various attack classes. This robust methodology will ensure the generation of a highly performing, generalizable IDS with the ability to precisely identify multiple cyber threats across different real-world network environments along with scalability and reliability.

The following algorithmic strategy outlines the step-by-step algorithm employed to build this research.

INPUT: CICIDS2018 dataset (CSV format) containing static and dynamic features of network traffic, labeled as benign or attack types.

1. **Import Necessary Libraries**

   Import all required Python libraries to support data preprocessing, deep learning, visualization, and model assessment. These libraries include NumPy and Pandas for data processing, Scikit-learn for preprocessing and Random Forest usage, TensorFlow and Keras for deep learning (CNN and LSTM), and Matplotlib/Seaborn for data visualization and performance plotting.

2. **Load and Preprocess Dataset**

   a) Import the CICIDS2018 dataset (namely, the Friday-02-03-2018_TrafficForML_CICFlowMeter.csv file) having labeled network traffic data.
   b) Eliminate unused columns like timestamps or flow IDs that won't be used for classification.
   c) Fill in missing values by performing imputation (e.g., with column mean).
   d) Convert categorical labels (e.g., "BENIGN", "DoS Hulk", etc.) to numerical form via Label Encoding.
   e) Normalize numerical features with z-score normalization for uniform feature scaling during training.

3. **Feature Reduction using PCA**

   a) Perform Principal Component Analysis (PCA) to dimensionally reduce the dataset while maintaining variance.
   b) Keep enough principal components (e.g., 95% variance cutoff) to strike a balance between model accuracy and computational costs.
   c) Project the dataset to its lower-dimensional PCA form for subsequent model training.

4. **Train Deep Learning Model:**

   a) CNN + LSTM Architecture Build a deep learning model that uses 1D Convolutional Neural Networks (CNN) for spatial feature learning and Long Short-Term Memory (LSTM) layers for temporal sequence analysis.
   b) Declare model layers such as Conv1D, LSTM, Dropout for regularization, and Dense layers for classification.
   c) Compile the model with the Adam optimizer and binary cross-entropy loss function.
   d) Train the model on the PCA-transformed training set and validate on held-out test data.
   e) Watch for training and validation accuracy and loss over epochs for convergence.

5. **Parallel Training of Random Forest Classifier**

   a) Initialize and train a Random Forest classifier on the same PCA-transformed dataset.
   b) Perform hyperparameter tuning on parameters like nu mber of trees and depth to tune for the best performance.
   c) Check its individual performance against classificatio n metrics to compare with other models.

6. **Ensemble Evaluation and Prediction**

   a) Ensemble the CNN+LSTM and Random Forest model predictions through majority voting.
   b) Make final predictions by comparing the outputs and taking the most confident or prevailing label.
   c) This ensemble process aids in improving detection strength and eliminating false positives.

7. **Model Evaluation and Visualization**

   a) Calculate and evaluate important evaluation measures like Accuracy, Precision, Recall, and F1-Score for the individual and ensemble models.
   b) Plot a confusion matrix to represent the classification performance for various categories of attacks.
   c) Plot Receiver Operating Characteristic (ROC) curves and compute Area Under Curve (AUC) to measure classification thresholds.

d) Plot training/validation loss and accuracy curves to test the generalization capability of the CNN+LSTM model.

OUTPUT: An optimized hybrid Intrusion Detection System (IDS) that effectively identifies malicious network behavior with diminished feature dimensions and sophisticated deep learning methods.

4.RESULT

This work employed a Hybrid Intrusion Detection System (IDS) was created in order to enhance the accuracy and efficiency of detecting malicious network traffic through the incorporation of both machine learning (ML) and deep learning (DL) approaches. The system was implemented using the CICIDS2018 dataset, which holds realistic network traffic data with several labeled cyberattack scenarios. The first process in the pipeline was data preprocessing, which included the removal of timestamp features, factorization-based encoding of categorical variables, missing and infinite value handling, and normalization via Standard Scaler. To address the high-dimensional character of the dataset and simplify computational complexity, Principal Component Analysis (PCA) was utilized for feature extraction, keeping the top 30 most informative components that account for most of the variance in the data. This phase not only achieved dimensionality reduction but also aided in the removal of redundant and less descriptive features, which may otherwise be a bottleneck to model performance.

Post-preprocessing, the data set was divided into training and testing sets in 80:20 proportion. In the deep learning part, a hybrid structure by integrating Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks was constructed. The CNN layer was tasked with extracting spatial features from the data, whereas the LSTM layers were used to capture sequential dependencies that are instrumental in discerning time-based attack patterns. This CNN+LSTM model was trained with the Adam optimizer and binary cross entropy loss function and was trained for more than 10 epochs with a batch size of 32. The model had very good learning capability as it was able to achieve a very high accuracy of 99.98% on the test set along with excellent generalization as reflected by the training and validation accuracy plot. The performance of the model was also evaluated using precision, recall, and F1-score metrics, and all of these measures reported high values, indicating efficient intrusion detection with low false positives and false negatives.

Besides the deep learning model, a Random Forest classifier was also trained on the same PCA-transformed dataset. Being known for its stability with structured tabular data and its resistance to overfitting, the Random Forest model showed a perfect accuracy of 100%. For tapping the strengths of both models, an ensemble approach was taken, aggregating the predictions made by the CNN+LSTM deep learning model and the Random Forest classifier. This ensemble methodology enabled the system to leverage the deep learning model's capacity for recognizing intricate patterns and sequences of data and take advantage of the high interpretability and dependability of the Random Forest model. The ensemble enhanced overall stability and detection resilience, especially in heterogeneous attack situations. The training and validation loss and accuracy is graphically displayed in Figure 1.

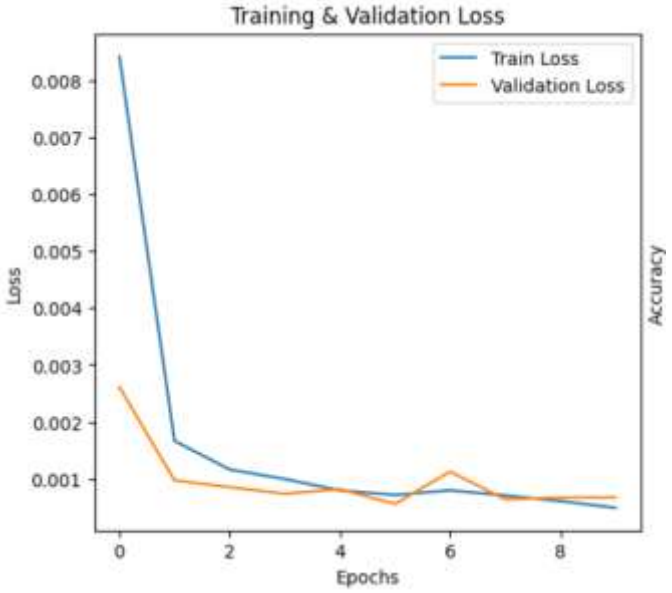

Figure 1(a): Training and validation loss of CNN+LSTM deep learning model for 10 epochs.

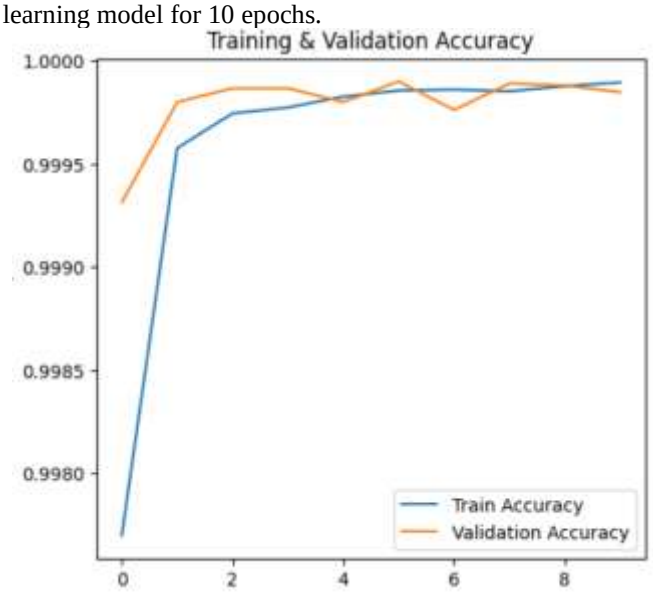

Figure 1(a): Training and validation accuracy of CNN+LSTM deep learning model for 10 epochs.

Both models were evaluated using accuracy, precision, recall, and weighted-average F1-score as metric, which is particularly vital in imbalanced classification problems. The CNN+LSTM model achieved a weighted-average F1-score of approximately 99.92%, and the Random Forest model achieved a flawless F1-score of 100%, which verified the system's excellent performance under various metrics. This holistic method successfully overcomes the shortcomings of conventional IDS by merging the virtues of ML and DL, hence providing a scalable, adaptive, and precise solution to contemporary network security. The final deep learning and random forest classifier classification report is presented in Figure 2

Classification Report - Deep Learning Model:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	152417
1	1.00	1.00	1.00	57298
accuracy			1.00	209715
macro avg	1.00	1.00	1.00	209715
weighted avg	1.00	1.00	1.00	209715

Figure 2(a): Classification report representing Deep Learning classifier metrics.

Classification Report - Random Forest Model:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	152417
1	1.00	1.00	1.00	57298
accuracy			1.00	209715
macro avg	1.00	1.00	1.00	209715
weighted avg	1.00	1.00	1.00	209715

Figure 2(b): Classification report representing Random Forest classifier metrics.

5. CONCLUSIONS AND FUTIRE SCOPE

The Hybrid Intrusion Detection System (IDS) methodology was carried out using PCA for dimensionality reduction, CNN+LSTM deep learning model for pattern detection, and Random Forest classifier for better accuracy and interpretability. The CNN+LSTM model was 99.98% accurate, whereas the Random Forest model was 100%, and the ensemble also increased detection resilience and removed false positives. Training and validation results confirmed strong learning and minimal overfitting. This hybrid approach resolves significant failings of traditional IDS, such as poor detection of emerging threats and high rates of false alarms, and is therefore a responsive and scalable solution for modern cybersecurity environments. The proposed hybrid Intrusion Detection System (IDS) exhibits high accuracy and resilience in a controlled environment, and it has great promise for real-world deployment in real-time applications. One of the main future prospects is to deploy this model within a real-time IDS system, where it can constantly observe live network traffic and raise alarms on identifying unauthorized intrusions. Combining the model with SIEM systems would improve threat correlation and automate response to incidents. Future development could also include adapting the system for IoT and edge environments through low-resource optimization. Further extension of the model to multi-class classification could be useful in determining particular types of attacks, allowing more in-depth threat analysis. Integration of transformer-based architectures and adversarial training methods could also further enhance detection of advanced threats. Lastly, having online learning would allow the system to continually adapt to emerging attack patterns, thus becoming more dynamic and robust in the long term.

6.REFERENCES

[1] A. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," 2015 International Conference on Signal Processing and Communication Engineering Systems (SPACES), Guntur, India, 2015, pp. 92–96. DOI: [10.1109/SPACES.2015.7058227].
[2] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," Journal of Computational Science, vol. 25, pp. 152–160, Sep. 2018.
[3] Y. Meidan et al., "Detection of Unauthorized IoT Devices Using Machine Learning Techniques," arXiv preprint arXiv:1709.04647, 2017.
[4] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," Future Generation Computer Systems, vol. 82, pp. 761–768, May 2018.
[5] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with LSTM recurrent neural networks," 2017 IEEE 33rd International Conference on Data Engineering Workshops (ICDEW), pp. 59–62.
[6] M. Sultana, N. Chilamkurti, and W. Peng, "Survey on SDN based network intrusion detection system using machine learning approaches," PeerJ Computer Science, vol. 4, e199, 2018.
[7] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Deep Learning Approach for Intelligent Intrusion Detection System," IEEE Access, vol. 7, pp. 41525–41550, 2019.
[8] S. Wu and Y. Zhang, "Network intrusion detection based on deep learning," 2020 15th International Conference on Computer Science & Education (ICCSE), pp. 55–60.
[9] R. Shahid, K. Ahmad, and M. M. Rathore, "Intrusion detection with deep learning using autoencoders and stacked LSTM," Future Generation Computer Systems, vol. 119, pp. 272–285, 2021.
[10] M. Mohammadi, F. Alazab, A. Jolfaei, and N. Kumar, "Internet of Things Malware Detection Using Convolutional Neural Network and Autoencoders," IEEE Transactions on Industrial Informatics, vol. 16, no. 11, pp. 7156–7165, Nov. 2020.
[11] Y. Kim, J. Kim, and H. Kim, "Enhancing Intrusion Detection Using Convolutional Neural Networks and Attention Mechanism," Sensors, vol. 20, no. 23, p. 6829, 2020.
[12] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," Journal of Network and Computer Applications, vol. 60, pp. 19–31, Jan. 2016.
[13] R. Roy, S. Biswas, and R. Islam, "A Deep Learning Based Hybrid Intrusion Detection System," 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), pp. 1–7.
[14] M. J. Hasan, M. S. Rahman, and M. M. Hasan, "A Hybrid Intrusion Detection Framework for Cloud Computing Using Feature Selection and Ensemble Learning," International Journal of Computer Applications, vol. 182, no. 46, pp. 1–9, 2021.
[15] S. Zarrin, S. Dehghantanha, and K. K. R. Choo, "Detecting botnets using collaborative deep learning," Future Generation Computer Systems, vol. 104, pp. 295–308, Mar. 2020.