International Research Journal of Multidisciplinary Scope (IRJMS), 2026; 7(1): 1499-1516 IRJMS

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2026.v07i01.07401

Ensemble Machine Learning Approaches to Strengthen Cyber Security and Network Defense

A Sivasangeetha[1], D Joseph Pushparaj[1], S Manjula[2], T Jasperline[3], R Saravanakumar[4]*

[1]Department of Computer Science and Engineering, PSN College of Engineering and Technology, Tirunelveli, Tamil Nadu, India, [2]Department of Computer Science and Engineering, Vel Tech Rangarajan Dr Sagunthala R & D Institute of Science and Technology (Deemed to be University), Chennai. Tamil Nadu, India, [3]Department of Computer Science and Engineering, Dr. G. U. Pope College of Engineering, Tuticorin, Tamil Nadu, India, [4]Iconix Software Solution, Tirunelveli, Tamil Nadu, India. *Corresponding Author's Email: iconixsaro@gmail.com

Abstract

For the protection of digital infrastructures, strong and flexible security structures are necessary due to the fast rise in cyber threats, such as spyware, phishing emails, and distributed denial-of-service (DDoS) attacks. By combining the advantages of several techniques, ensemble machine learning (EML) has become a potent paradigm to improve cyber defense by increasing detection accuracy and resistance against changing attack vectors. In order to successfully detect and prevent network intrusions, this research investigates an ensemble strategy that makes use of K-Nearest Neighbors (KNN), Long Short-Term Memory (LSTM) networks, and Multi-Layer Perception (MLP) models. MLP offers nonlinear feature training for complicated threat landscapes, LSTM is excellent at identifying sequential relationships in network data, and KNN offers effective recognition of patterns for static attack signatures. By combining these models, temporal and geographical features are exploited, lowering false positives and improving prediction accuracy. Recent benchmark datasets, such as CIC-DDoS2019, are used to assess performance in a variety of attack scenarios, offering a thorough understanding of practical application. The suggested ensemble performs noticeably better than individual models in accuracy, precision, and recall, according to experimental data, making it a viable instrument for proactive cyber defense tactics. This study emphasizes how ensemble learning may improve cybersecurity and network resilience in a revolutionary way.

Keywords: Cybersecurity, Distributed Denial-Of-Service, Ensemble Machine Learning, K-Nearest Neighbors, Long Short-Term Memory Networks, Multi-Layer Perception.

Introduction

There has been a meteoric rise in the use of web-based applications and services within the last two decades. As of right now, 57% of the global population is online. Because of this, worries about the safety of the internet have grown substantially. Numerous security threats have often been present on the Internet. Online anomalies such as Trojan horses, malware, port scanning, and DoS attacks are commonplace (1). When dealing with large and complicated networks, typical network topologies often fail. An alternative method that uses software rather than hardware components like switches and routers to manage network traffic is known as software-defined networking (SDN). A centralized controller acts as the principal decision-maker for the network in SDN, taking over the control plane (2-3).

In response to these limitations, researchers have used advanced machine learning and deep learning approaches, particularly neural networks, to create context-aware forecasting systems for detecting DDoS attacks and predicting. However, existing methods in the literature employ outdated datasets for training and struggle to distinguish between legitimate traffic and application-layer DDoS attacks (4).

The potential of EML to develop more reliable and accurate attack detection systems has garnered a lot of research interest. This method overcomes the drawbacks of individual algorithms, such as bias, overfitting, and inadequate generalization of fresh data. Ensemble learning continuously outperforms individual models by merging many classifiers, leading to improved accuracy. To enhance model performance, a variety of ensemble algorithm combination techniques, including bagging, stacking, and boosting, may be set up and created. It is also crucial to use the capabilities and

(Received 04th August 2025; Accepted 30th December 2025; Published 31st January 2026)

advantages of distinct algorithms in various contexts. In various settings with varying dimensionalities, machine learning algorithms provide a unique collection of features and operational efficiency. By combining these accessible capabilities, ensemble learning creates a system that can accurately detect possible unknown attacks. Furthermore, enabling the efficiency advantages of another method might help prevent the drawbacks of a certain approach. A benchmark dataset like CIC-DDoS2019, which includes a variety of characteristics pertaining to network traffic and intrusion detection, is used to guarantee a reliable experimental assessment. To increase overall prediction accuracy, decrease computing complexity, and improve detection performance, the dataset characteristics are examined, assessed, and processed.

On the other hand, passive attacks include secretly watching data transfers to get private information or unencrypted passwords (5–7). These attacks don't change system resources, yet they are nonetheless quite dangerous since they break data privacy without anybody knowing. Figure 1 shows common threats such as DoS and DDoS attacks, Man-in-the-Middle attacks, packet sniffing, port scanning, and ARP spoofing (8-9).

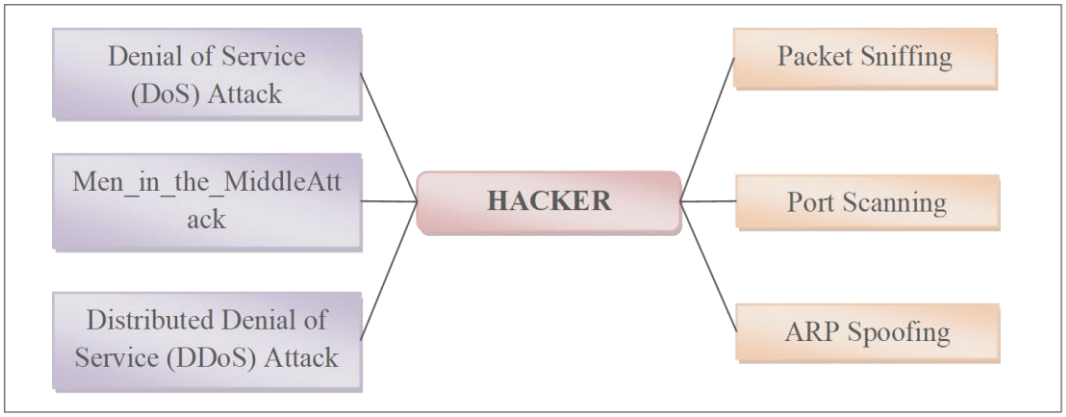


Figure 1: Cyberattack Architecture

Threat Model

Large-scale and varied DDoS attack characteristics seen in the CIC-DDoS2019 dataset, such as high-rate flooding assaults, protocol exploitation, and temporally scattered attack patterns, are now explicitly taken into account by the Threat Model. Additionally, it takes into consideration adversaries trying to avoid detection by imitating legal traffic, which directly drives the employment of neighborhood-based (KNN), nonlinear (MLP), and temporal (LSTM) learning processes.

Background

The exponential expansion of digital connections and the increasing number of smart gadgets have increased the potential attack surface for hackers. Threats, including large-scale DDoS assaults, zero-day vulnerabilities, and advanced persistent threats (APTs), have therefore become more complex. Conventional security methods that rely on rules and signatures work well against known threats, but they can't keep up with the constantly evolving nature of modern attacks. Machine learning techniques have gained popularity in cybersecurity for identifying irregularities and forecasting dangers because of their data-driven and flexible nature. However, because of problems like excessive fitting, skewed learning, and poor generalization, relying only on one machine learning model may be troublesome when confronted with novel attack patterns. To overcome these challenges, EML techniques that integrate many models, such as MLP, KNN, and LSTM, have been successful. Ensemble techniques use the advantages of many algorithms to enhance detection precision, false positive rate, and network defensive capabilities. Using current benchmark datasets like CIC-DDoS2019, researchers can train and evaluate robust hybrid approaches that change with dynamic cyber environments, enhancing the adaptability of critical systems against known and unknown attacks.

Motivation

a) Modern digital infrastructures, such as cloud platforms, IoT networks, and business systems, are seriously threatened by the sharp growth in sophisticated cyberattacks like DDoS, phishing, and malware.
b) High false alarm rates and poor generalization are often the consequence of traditional

security measures and individual machine learning models' inability to adjust to changing attack patterns.

c) A single learning model is unable to adequately represent the temporal interdependence and static features of network traffic.
d) Combining complementary classifiers to increase detection accuracy and resilience is made possible by recent developments in ensemble machine learning.
e) Advanced intrusion detection systems may be realistically evaluated under a variety of assault scenarios thanks to benchmark datasets like CIC-DDoS2019.
f) Scalable, precise, and intelligent cybersecurity frameworks that can function dependably in fast and diverse network contexts are desperately needed.

Problem Statement

Even with state-of-the-art intrusion detection systems (IDS), it is very challenging to accurately detect new or developing cyber threats. Traditional IDSs rely on pattern recognition, which makes them ill-equipped to detect sophisticated attacks such as zero-day vulnerabilities. Nevertheless, IDS that build detection models using particular ML techniques have a lot of flaws, such as bias, over-fitting, and poor data standardization. The implementation of these constraints raises the possibility of false positives, in which legitimate traffic is incorrectly identified as an attack, and false negatives, in which attacks are not detected. EML is a practical approach to increasing the accuracy of attack detection while overcoming these limitations. When it comes to NIDS, it's important to study the pros and cons of different ensemble techniques and algorithm combinations in terms of comprehensibility, complexity, and computational power. A machine learning-based intrusion detection system that minimizes false positives and false negatives while precisely recognizing and classifying various DDoS attack types from high-dimensional and time-dependent network traffic. The updated definition also highlights issues with class imbalance, real-time detection restrictions, and traffic fluctuation.

Research Gap

a) Single classifiers, which are inadequate for managing intricate and dynamic cyberattack behaviors, are the basis of the majority of current intrusion detection research.
b) Hybrid ensemble frameworks that combine temporal (LSTM), distance-based (KNN), and nonlinear (MLP) learning models have not received much attention.
c) Current ensemble approaches often concentrate only on accuracy, failing to adequately analyze temporal attack patterns, false positives, and false negatives.
d) A lot of research fails to properly assess ensemble models using current, actual datasets like CIC-DDoS2019.
e) In ensemble-based cybersecurity research, scalability, computing cost, and real-time deployment issues are often disregarded.
f) The limits and practical application of ensemble learning in actual network protection systems are not well discussed.

Objective

This study's goal is to identify a variety of cyberthreats by integrating the complementary capabilities of KNN, LSTM, and MLP in an ensemble machine learning framework. By lowering false positives and raising accuracy, recall, and F1-score, the suggested model seeks to outperform individual classifiers. Modern benchmark datasets like CIC-DDoS2019 are used to assess their efficacy and guarantee dependable performance in current attack situations. The ensemble is appropriate for real-time network monitoring and security since it was created with low-latency prediction in mind. The system successfully handles changing attack vectors by combining the sequential learning capabilities of LSTM, the nonlinear feature abstraction of MLP, and the pattern recognition of KNN. In order to offer scalable and reliable cybersecurity defense, the paper also provides a practical roadmap for incorporating the ensemble model into intrusion detection systems (IDSs) and security operations centers (SOCs). Below is the literature work.

This approach quickly and accurately identified botnets while using few resources. Negative aspects include a low recognition rate, high complexity, and unpredictability. Presented here was a hybrid technique for selecting characteristics and categorizing cyberattacks. The k-means clustering technique and the correlation-based selection of features method were combined to produce an ideal feature subset. In order to do classification, the decision tree (J48) was merged with the stochastic Naïve Bayes (NB) method. The

complexity of its structure, along with its high false-positive rates, is a drawback (10). A botnet traffic analyzer called BoTShark was developed using deep learning. This method got over the restriction on using encrypted payloads. Another interesting finding is that there were correlations between the original and recovered attributes discovered by every Convolutional Neural Network (CNN) layer. The Softmax classification algorithm was used as an indicator to successfully detect fraudulent traffic (11). Learning techniques to ensure that people comprehend IoT, as well as having a comprehensive awareness of various malware and how to detect them, are the primary focus of the study (12). Then, with an emphasis on deep learning approaches, we examined and evaluated the current status of IDS research in four main datasets (13).

Numerous studies have focused on cyberattack predictions using machine learning techniques, such as ensemble methods. Network traffic categorization has made substantial use of supervised machine learning algorithms. Training a Network Intrusion Detection System's (NIDS) rule-based model on a variety of datasets may result in greater accuracy in classification and lower false positive rates. On the other hand, unsupervised machine learning techniques have been used to evaluate connections and clustering methods in network data, which might aid in the identification of previously unidentified attack patterns. However, the main focus of this study is on supervised machine learning techniques based on binary classification (14-15).

A flexible NIDS EML model. Model weights are dynamically adjustable, and particular model configurations are also dynamically adjusted. Multiple decision trees, k-NN, DNN, and random forests are used as base classifiers. Averaging is a method of adaptive voting. The proposed architecture achieved an accuracy of 84.2% when evaluated on the NSL-KDD dataset, while an adaptive voting mechanism achieved an accuracy of 85.2%. For improved outcomes, the authors suggest making the most of feature selection and preprocessing (16).

In IDSs, machine learning is much more effective than conventional approaches, especially in light of the increasing complexity of network threats (17). System administrators were tasked with manually searching logs for faults according to Jim Anderson's 1980 notion (18). Conversely, cutting-edge IDS systems are making more and more use of intelligent automation that makes use of machine learning methods (19).

This study significantly enhances the capability to collect real-time network data, which serves as input for the identification of anomalies algorithm. The structure and behavior of data flows across networks have evolved a lot because networking technology is always becoming better. The significance of recording network traffic precisely at the moment of intrusions, particularly during data exchanges between systems, has increased in recent years due to advancements within communication protocols and dissemination methods. To find bad activity, it's important to capture network data rapidly and precisely during these periods of transmission (20).

Machine learning and ensemble techniques for improving cybersecurity and intrusion detection systems has been the subject of several studies. With an emphasis on their flexibility and efficacy in complex network contexts, recent research offers a comprehensive review of machine learning-driven methods for identifying and reducing a variety of cyberthreats (21). By showing that integrating many models may greatly lower false positives and increase detection accuracy, the improved ensemble defense framework was presented to increase the adversarial resilience of intrusion detection systems (22). A hybrid machine learning approach that improves cyberattack detection in cloud computing infrastructures by using both supervised and unsupervised techniques (23). The use of machine learning in financial systems has also been studied, emphasizing its function in maintaining cybersecurity in digital banking platforms via real-time threat prediction and mitigation (24). The combined use of several classifiers surpasses single-model techniques in terms of accuracy and resilience by presenting an ensemble framework that can identify and classify cyberattacks successfully. Together, these studies demonstrate the increasing significance and efficacy of ensemble and hybrid machine learning approaches for reliable and expandable cybersecurity solutions (25).

Methodology

This section describes in detail our methodology for detecting and classifying DDoS attacks in the CIC-DDoS2019 dataset. Incorporating LSTM and EML into a KNN and MLP model is our approach. The LSTM module enables the analysis of complex and massive datasets, and the LSTM model employs a probabilistic strategy that accounts for the uncertainties and probabilities associated with network traffic patterns. The first stages include gathering, cleaning, and preparing the data. Principal Component Analysis (PCA) is used to decrease dimensionality after the extraction of relevant information. We proceeded to build and train the EML classifier, verify the model, and evaluate it using pertinent metrics such as F1 Score, recall, accuracy, and precision. Finally, we save the data from the trained model. Data Fusion is included to assess uncertainty; the LSTM method integrates features extracted from many data sources. Findings from the newly trained model are contrasted with those from the EML model and other similar models. Finally, we propose an approach that relies on an EML model to detect DDoS attacks.

The EDA phase involves tasks such as data visualization, feature generation, and correlation assessment to ensure high-quality data for training models and to understand the dataset. After EDA, the processed attributes are kept, and a variety of ML algorithms are used. By running the model through an accuracy check, we can find out whether its predictions are satisfactory. If that doesn't work, the approach follows the usual cycle of improving machine learning model development by recommending algorithm and characteristic adjustments until the desired accuracy is reached. As seen in Figure 2, the research study's procedure includes data preparation, model building, ensemble learning, real-time implementation, and clarity integration.

Cyberattacks, which put users' data security and privacy at risk, have been on the increase as we rely more on equipment that is connected to the internet. To protect networks from unwanted access, several security measures have been put in place, including firewalls, IDS, and anti-malware software. These systems can do anything from basic rule matching to complex intelligent models.

CIC-DDoS2019 Dataset

An essential starting point for assessing the efficacy of EML techniques meant to improve cybersecurity and the defense of networks is the CIC-DDoS2019 dataset. The data, which was created by the Canadian Institute for Cybersecurity, includes more than 50 different kinds of DDoS assaults that are categorized into HTTP, UDP, and TCP-based floods, as well as amplification attacks that closely resemble actual situations. Robust model training and testing are made possible by its architecture, which incorporates both benign and malicious traffic recorded via realistic testbed settings. The dataset is suitable for a variety of detection techniques as it offers packet-level captures (PCAP) files and flow-based characteristics, including packet length, length of time, and inter-arrival periods. CIC-DDoS2019 provides a rich and extremely dimensional feature space for ensemble learning models (ELM), such as those that combine KNN, LSTM, and MLP, allowing them to take advantage of the advantages of many algorithms at once. It fills in the gaps left by previous datasets and aids in the creation of flexible, highly accurate IDS by recording complex and changing assault patterns. The assessment of the suggested ensemble architecture in this research is supported by CIC-DDoS2019, which guarantees its applicability to contemporary cyber threat environments and increases its capacity for proactive network security and real-time DDoS detection.

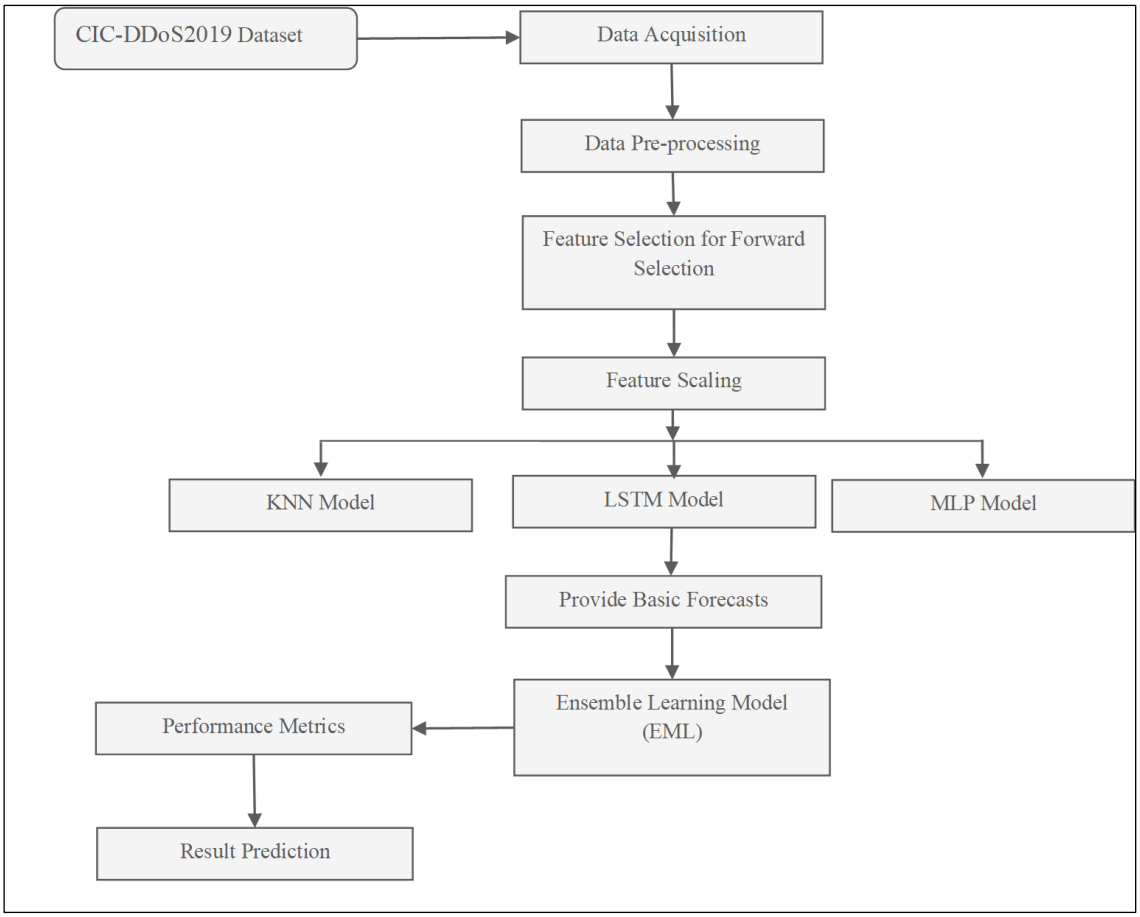


Figure 2: System Architecture

Data Acquisition

The acquisition of varied, high-quality datasets that capture both benign and harmful network activities is the basis of the suggested EML Approaches to Strengthen Cyber Security and Network Defense. To guarantee accuracy and applicability, benchmark datasets like CIC-DDoS2019 were obtained for this work from respectable cybersecurity research organizations. These datasets provide a balanced perspective for model training and assessment by combining genuine traffic with a variety of contemporary cyberthreats, such as DDoS, brute-force, botnet, and reconnaissance assaults. Gathering packet-level captures (PCAP files) and turning them into flow-based feature sets with characteristics like packet size, protocol type, flow length, and inter-arrival periods were all part of the data-gathering process. To avoid bias in the learning process, this preparation step also included addressing missing values, cleaning and normalizing the data, and class balancing. The ensemble approach gains from diverse traffic patterns by combining several datasets, which helps it generalize well across a range of scenarios for attacks and network infrastructures.

Data Pre-Processing

A thorough data pre-processing pipeline was put in place before model training in order to guarantee the dependability and effectiveness of the EML Approaches to Strengthen Cyber Security and Network Defense. To eliminate duplicate entries, unnecessary characteristics, and missing records, raw network traffic data from benchmark datasets like CIC-DDoS2019 was cleaned. While categorical data, such as protocol kinds, were converted using one-hot encoding to make them compatible with machine learning models, missing values were handled using imputation or elimination. To guarantee uniform scaling, which is essential for algorithms like KNN and MLP that are susceptible to feature size, continuous features were normalized or standardized. To eliminate bias toward majority classes, oversampling Synthetic Minority Oversampling Technique (SMOTE) and under-sampling approaches were used to alleviate class imbalance, a typical problem in cybersecurity datasets. Following preprocessing, the dataset was

methodically separated into subsets for testing (15%), validation (15%), and training 70%. The testing set offered an objective assessment of the ensemble's performance, the validation set adjusted hyperparameters and avoided overfitting, and the training set made it easier to learn the model. The ensemble framework was trained on clean, balanced, and representative data thanks to this organized pre-processing and data split-up technique, which produced more accurate and broadly applicable cyber threat identification findings.

Feature Selection for Forward Selection

This study used Forward Selection on the CIC-DDoS2019 dataset to systematically determine the most important network traffic variables for the EML framework. All of the more than 80 flow-based parameters included in the CIC-DDoS2019 dataset, including packet size, flow duration, protocol flags, and inter-arrival delays, might be overwhelming for model training when used in their entirety. When using Forward Selection, the ensemble model is trained on each characteristic, such as packet size, flow time, or protocol type, independently; this is done starting with no features. Here, the CIC-DDoS2019 dataset is used. To begin, we provide the feature (accuracy, F1-score, etc.) that has the greatest impact on enhancing detection performance. The process then iteratively continues, this time adding to the subset the characteristics that provide the highest performance increase following each other's evaluation alongside the previously selected features. This approach continues until either the number of features reaches a certain limit or adding additional features no longer improves the model significantly. Since only the most important and useful properties from CIC-DDoS2019 are selected, noise is decreased without compromising critical indicators for detecting DDoS attacks.

Feature Scaling

To provide consistent input for the ensemble framework, the CIC-DDoS2019 dataset employs feature scaling to normalize numerical parameters such as packet size, flow duration, and byte counts. Because algorithms like KNN and MLP are very sensitive to changes in feature magnitudes, standardization, which involves converting data to have zero mean and unit variance, and Min-Max normalization, which involves scaling values between 0 and 1, were used. Because this preprocessing step ensures that no single feature dominates the learning process due to its size, the ensemble of KNN, LSTM, and MLP can train more effectively, achieve faster convergence, and deliver balanced, high-accuracy detection of DDoS attacks.

Data Classification

Data classification in the proposed EML framework makes use of a combination of KNN, Multi-Layer Perceptron (MLP), and LSTM models, all of which have their distinct analytical capabilities. By classifying network traffic according to the distance between a new data point and its closest neighbors in the feature space, KNN, acting as a lazy learner, helps identify known attack patterns in static data. By analyzing the normalized and selected data over several hidden layers, MLP, a deep feed-forward neural network, captures complex, non-linear connections between malicious and benign traffic characteristics. Meanwhile, LSTM, a kind of recurrent neural network (RNN), can simulate sequential dependencies in network flow data, making it possible to identify time-dependent attack fingerprints like evolving DDoS patterns. To improve the accuracy and reliability of classifying network traffic as either legal or malicious, the ensemble framework combines the predictions of various classifiers. This is achieved by merging LSTM's temporal learning capabilities with KNN's pattern recognition and MLP's feature abstraction. Overall, cybersecurity is enhanced, false positives are reduced, and detection accuracy is increased by this cooperative technique.

K-Nearest Neighbors (KNN)

To detect hostile and benign traffic patterns in the CIC-DDoS2019 dataset, the K-Nearest Neighbors (KNN) method is used as a baseline classifier inside the suggested Machine Learning Approaches to Strengthen Cyber Security and Network Defense. Using parameters like packet size, flow time, and byte count, KNN compares a new network flow to its k closest neighbors in the training data to classify it, as shown in Table 1. KNN is a distance-based, non-parametric technique. In CIC-DDoS2019, KNN was able to identify several types of DDoS attacks, such as UDP floods, SYN floods, and amplification assaults, by using metrics like Euclidean distance to identify commonalities in traffic flow. Due to KNN's lack of data distribution assumptions, it can adapt well to this dataset's

diversified and high-dimensional feature space. To avoid computational inefficiencies and bias caused by features with wider numerical ranges, preprocessing procedures like feature scaling and dimensionality reduction were essential.

Table 1: KNN Pseudocode

Step 1: Load the Dataset
Import the CIC-DDoS2019 dataset.
Load flow-based features (packet size, flow duration, protocol flags) and labels (benign or DDoS attack types).
Step 2: Pre-process Data
Handle missing values (remove or impute).
Encode categorical features (e.g., protocol type → one-hot encoding).
Apply feature scaling (Min-Max or Standardization).
Split the dataset into a training set (70%) and a testing set (30%).
Step 3: Choose Parameters
Select k (number of neighbors, k=5).
Choose a distance metric (Euclidean, Manhattan, or Minkowski).
Step 4: Training Phase (Lazy Learning)
Store all training data points and their labels in memory (KNN does not "train" in the traditional sense).
Step 5: Classification Phase (For each test sample)
a. Calculate the distance between the test sample and all training samples.
b. Sort distances in ascending order.
c. Select the k nearest neighbors.
d. Count the labels of these k neighbors (benign vs. attack types).
e. Assign the majority label to the test sample.
Step 6: Evaluation
Compare predicted labels with actual labels.
Compute performance metrics: Accuracy, Precision, Recall, F1-score.

Multi-Layer Perceptron (MLP)

To classify network traffic from the CIC-DDoS2019 dataset, the proposed Machine Learning Approaches to Strengthen Cyber Security and Network Defense mainly depend on the MLP for learning complex, non-linear connections between benign and malicious flows. The dataset's diverse properties, such as packet length, flow duration, and inter-arrival periods, are entered in the MLP's input layer, shown in Table 2. To find deeper patterns that could point to DDoS attacks, many hidden layers use weighted transformations and non-linear activation functions, such as ReLU. The MLP continuously modifies weights during training via backpropagation, lowering the classification error, with optimizers such as Adam. After training, the network can accurately identify freshly arriving traffic records as either benign, UDP flood, or SYN flood. Because of the high dimensionality of the data it contains, CIC-DDoS2019 required preprocessing techniques, including feature scaling and forward selection, to improve convergence and reduce overfitting.

Table 2: MLP Pseudocode

Step 1: Load the Dataset
Import the CIC-DDoS2019 dataset.
Extract features (X) such as packet size, flow duration, and byte counts.
Extract labels (y) (benign, UDP flood, SYN flood, etc.).
Step 2: Data Pre-processing
Handle missing values (impute or remove).
Encode categorical features (e.g., protocol type → one-hot encoding).
Scale features using Min-Max normalization or Standardization.
Split the dataset into a Training set (70%), Validation set (15%), and Testing set (15%).
Step 3: Initialize MLP Architecture
Define Input Layer: Number of neurons = number of selected features.
Add Hidden Layers (1–3 layers, with ReLU activation).
Add Output Layer:
Softmax activation (for multi-class attack classification)
Sigmoid activation (if binary classification: benign vs. attack).
Step 5: Set Training Parameters
Loss function: Cross-Entropy Loss (for multi-class tasks).
Optimizer: Adam (adaptive learning).
Define the number of epochs and batch size.
Step 6: Training Phase
For each epoch:
a. Feed training data through the network (forward pass).
b. Compute the prediction error (loss).
c. Perform backpropagation to update weights using gradients.
d. Adjust weights with the optimizer to minimize loss.
Step 7: Validation Phase
After each epoch, evaluate the model on the validation set to monitor overfitting and adjust hyperparameters if needed.
Step 8: Testing Phase

Run the trained MLP on unseen test data. Generate predicted labels. **Step 9:** Performance Evaluation Compute metrics: Accuracy, Precision, Recall, F1-Score. Analyze the confusion matrix to identify misclassifications.

Long Short-Term Memory (LSTM)

The LSTM network is used in the proposed Machine Learning Approaches to Strengthen Cyber Security and Network Defense to capture the temporal and sequential patterns found in the CIC-DDoS2019 dataset, which comprises flow-based records of both malicious and benign traffic. In contrast to conventional feed-forward networks, LSTM is a specific kind of RNN that is very successful in analyzing time-series network traffic data because it is designed to manage long-term dependencies and solve the vanishing gradient issue shown in Table 3. In the LSTM, memory cells use input, output, and forget gates to selectively keep or delete information based on features including packet arrival timings, flow lengths, and inter-packet intervals. Through this method, the model may discover patterns that static models would miss by learning how DDoS attack behaviors change over time. The LSTM provides rich contextual knowledge into attack evolution by classifying traffic flow sequences by categories such as benign, SYN flood, or UDP flood when applied to CIC-DDoS2019. When included in the ensemble, LSTM has strong temporal learning capabilities that enhance the instance-based recognition of KNN and the non-linear feature abstraction of MLP, resulting in a comprehensive and flexible cybersecurity defensive model.

Table 3: LSTM Pseudocode

Step 1: Load the Dataset Import the CIC-DDoS2019 dataset. Extract features (X) such as packet size, flow duration, and inter-arrival times. Extract labels (y) (benign, UDP flood, SYN flood, etc.). **Step 2:** Pre-processing Handle missing or inconsistent values. Encode categorical data (protocol types → one-hot encoding). Apply feature scaling (Min-Max or Standardization). Reshape the feature data into a 3D format: (samples, timesteps, features) for LSTM input. Split the dataset into training (70%), validation (15%), and testing (15%) sets. **Step 3:** Build an LSTM Model Define Input Layer: matching the shape of (timesteps, features). Add one or more LSTM layers with memory cells (64 or 128 units). Optionally include Dropout layers to prevent overfitting. Add a Dense (Fully Connected) layer for classification. Use Softmax activation for multi-class classification or Sigmoid for binary classification. **Step 4:** Compile Model Set loss function: Categorical Cross-Entropy for multi-class Binary Cross-Entropy for binary classification Choose optimizer: Adam (common choice for LSTM). Define evaluation metrics (accuracy, precision, recall). **Step 5:** Train Model For each epoch: a. Feed training sequences into the LSTM (forward pass). b. Calculate the loss and generate gradients (backpropagation through time). c. Update weights using the optimizer. d. Validate on the validation set to track performance and prevent overfitting. **Step 6:** Test Model Use the testing set for unseen traffic data. Generate predictions (benign or attack class labels). **Step 7:** Evaluate Performance Compute metrics: Accuracy, Precision, Recall, F1-Score. Analyze the confusion matrix to identify misclassifications.

Basic Forecasts

A first assessment of the effectiveness and suitability of the suggested EML Approaches to Strengthen Cyber Security and Network Defense is given via basic predictions. Using classifiers like KNN, MLP, and LSTM on the CIC-DDoS2019 dataset, the ensemble should minimize false positives and achieve high detection accuracy. According to preliminary estimates, the system will be able to successfully adjust to a variety of DDoS assault patterns and changing threat landscapes thanks to the combination of instance-based learning KNN, deep feature extraction MLP, and temporal sequence modeling LSTM. The predictions also point to increased preparedness for real-time intrusion detection applications, less

computing cost via optimal feature selection, and better generalization across various network settings. These projections provide a solid performance baseline, but they will be verified by thorough testing and comparison with current cybersecurity models.

Ensemble Learning Model (ELM)

An ELM combines the advantages of KNN, MLP, and LSTM in the suggested EMLApproaches to Strengthen Cyber Security and Network Defense to categorize network traffic in the CIC-DDoS2019 dataset more accurately and robustly than any one model could show in Table 4. A unique capacity is contributed by each base learner: LSTM captures temporal correlations in sequential traffic flows, MLP recovers intricate, non-linear feature interactions, and KNN provides robust instance-based detection of static attack signatures. A voting system or weighted average technique is used to aggregate their separate forecasts, resulting in a single categorization output for both benign and malevolent classes, including amplification assaults, UDP floods, and SYN floods. While enhancing their combined capabilities, this hybrid technique lessens the drawbacks of individual models, such as LSTM's computational expense or KNN's sensitivity to high-dimensional data. The ensemble model delivers strong detection performance, enhanced generalization across a variety of DDoS attack types, and reduced false-positive rates by using the rich flow-based properties of CIC-DDoS2019. This makes it a potent and scalable solution for contemporary network protection.

Table 4: ELM Pseudocode

Step 1: Load the Dataset
Import the CIC-DDoS2019 dataset.
Extract features (X) (packet size, flow duration, byte counts).
Extract labels (y) (benign, UDP flood, SYN flood, etc.).
Step 2: Data Pre-processing
Handle missing values (impute or remove).
Encode categorical features (protocol type → one-hot encoding).
Apply feature scaling (Min-Max or Standardization).
Optionally perform feature selection (Forward Selection) to remove irrelevant features.
Split data into Training (70%), Validation (15%), and Testing (15%) sets.
Step 3: Initialize Base Classifiers
Define the KNN model with the chosen k value and distance metric.
Define an MLP model with input, hidden, and output layers.
Define an LSTM model with a time-series input structure.
Step 4: Train Base Models
For each model:
a. Train KNN using the training set (store data for distance calculation).
b. Train MLP with forward pass and backpropagation.
c. Train LSTM with sequential data (using backpropagation through time).
Step 5: Generate Predictions (Validation and Test Sets)
Feed validation/test samples into KNN, MLP, and LSTM.
Collect each model's predictions for every sample.
Step 6: Combine Predictions (Ensemble Step)
Apply majority voting (for classification) or weighted voting (if accuracy weights are assigned to models).
Generate final class label (benign or attack category).
Step 7: Evaluate Ensemble Model
Compare ensemble predictions with true labels.
Compute metrics: Accuracy, Precision, Recall, F1-Score.
Analyze the confusion matrix to assess misclassification patterns.
Step 8: Output Results
Report final detection performance.
Highlight the ensemble's improvement over individual models.

Performance Metrics

In the classification trials, we evaluate the performance of our models using many critical metrics:

In order to quantify the efficacy of machine learning models, indicators of assessment are essential. F1 Score, recall, accuracy, and precision are some of the most common evaluation metrics used in model classifications.

Accuracy

Equation [1] describes the accuracy in cybersecurity, which refers to the completeness of a detection system, such as an intrusion detection or malware classification tool. It displays the frequency with which the model separates harmful from secure network traffic. A high accuracy score, for example, would mean that the system is correctly classifying most incoming data as either safe or dangerous in the framework of smart homeowner network monitoring. However, if the

dataset is uneven, accuracy may be misleading in real-world cybersecurity scenarios. For instance, if the majority of traffic is routine and just a small percentage represents an attack, the system may seem correct, yet overlook significant dangers.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad [1]$$

Where TP is the True Positive, correctly predicted the positive cases, TN is the True Negative, correctly predicted the negative cases, FP is the False Positive, incorrectly predicted the positive cases, and FN is the False Negative, incorrectly predicted the negative cases.

Precision

Equation [2] describes detecting something as malicious, and cybersecurity precision measures how reliable the system's alerts are. When a very precise system warns of a potentially harmful file, email, or connection, it is likely correct; such notifications are very rare from such a system. This is especially important in domains where a large volume of false positives may overwhelm security teams or cause users to ignore alerts, such as phishing detection, intrusion detection, and spam filters. By making sure that cybersecurity experts are confident in the signals that they receive, high accuracy lowers the time wasted on innocent behavior that is falsely categorized as attacks.

$$Precision = \frac{TP}{TP + FP} \quad [2]$$

Recall

Equation [3] describes the Recall in cybersecurity, which refers to a system's ability to recognize real threats. For example, a high-recall DDoS attack detection appliance efficiently detects most of the malicious traffic trying to overload the network. On the other hand, a low recall rate suggests that the infrastructure is missing a large number of genuine threats, thus leaving the network vulnerable. It is important for security-sensitive applications like fraud prevention or malware detection because, even if the system is accurate most of the time, failing to identify an attack might have disastrous consequences.

$$Recall = \frac{TP}{TP + FN} \quad [3]$$

F1-Score

A cybersecurity simulation's efficacy may be fairly evaluated using the F1-Score, which combines accuracy and recall into one score. Unfortunately, cybersecurity teams frequently face the difficult choice between tuning their systems for high accuracy and designing them for high recall. The former may miss some attacks to avoid unnecessary warnings, while the latter may detect every possible danger but generate an elevated number of false alarms. Equation [4] describes the F1-Score shines in scenarios where detecting threats and avoiding unnecessary false positives are of equal importance, such as anomaly surveillance in smart homes or intrusion detection. Theoretically sound and practically applicable to real-world protection, it helps ensure that the entire structure is well-designed.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad [4]$$

Result Prediction

The proposed ensemble model, which integrates KNN, MLP, and LSTM, is expected to outperform the individual models in terms of classification accuracy on the CIC-DDoS2019 dataset by leveraging their distinct strengths. By combining the instance-based recognition of KNN with the deep feature abstraction of MLP and the temporal sequence learning of LSTM, we anticipate that the ensemble will enhance the detection of different forms of DDoS attacks and decrease false positives. With the use of Forward Selection and feature scaling, a more effective model may converge

faster and generalize better to unexpected network data. Overall, the anticipated results demonstrate that the ensemble framework will provide a reliable cybersecurity system that can adapt to the network's demands and resist emerging and evolving threats. All three models are beneficial to the ensemble, but LSTM is expected to be the most significant contribution, boosting the total detection capabilities and guaranteeing that the framework can more accurately adjust to contemporary, dependent on time security threats.

Scalability

The proposed ensemble approach is meant to be scalable to edge, fog, and cloud layers, among other multi-layer network topologies. While the LSTM model may be run at higher layers (fog/cloud) to examine temporal traffic patterns, lightweight components like KNN and MLP can be deployed at edge or fog nodes for quick initial identification. The system can manage growing network capacity and traffic volume thanks to this tiered implementation without experiencing appreciable performance deterioration.

Table 5: Experimental Configuration List

Component	Configuration
Processor (CPU)	Intel Core i7.5 GHZ
Graphics Card (GPU)	NVIDIA 2GB
RAM	32GB DDR4
IDE	Jupyter Notebook
Software	Anaconda Python
Libraries	Scikit-learn, Keras, TensorFlow, Pandas, Numpy, Matplotlib

Results and Discussion

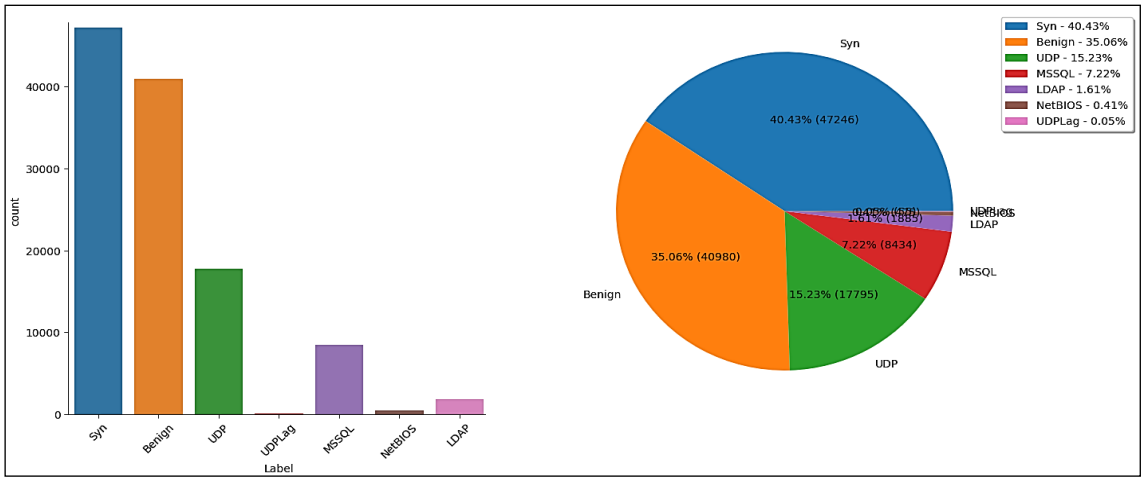
The experience assessment was conducted on a personal computer that met the specifications given in Table 5, above.

A thorough analysis of computational cost has been included, emphasizing the trade-off between higher processing overhead and better detection performance. The complementary characteristics of KNN, LSTM, and MLP greatly minimize false positives and false negatives, even if ensemble learning needs more computing than single models, owing to simultaneous training and inference. According to the article, layer-wise deployment and model optimization may reduce the additional computational cost, which is warranted for security-critical applications.

Label distribution and frequency in the CIC-DDoS2019 dataset highlight the need for comprehensive data cleaning and preprocessing before model training. At first, the dataset included both harmless and malicious traffic, with the former comprising 40.43%, the latter 35.06%, UDP 15.23%, MSSQL 7.22%, and minor amounts of LDAP, NetBIOS, and UDPLag. The data was prepared by removing duplicate and malformed items and handling missing values consistently. Methods like resampling and class balancing were investigated to make sure those rare attack types, which only contribute tiny percentages (e.g., UDPLag at 0.05%), shown in Figure 3 (A) and (B), did not affect model learning. To make features like packet size and flow time more similar, we standardized or normalized the feature values. These helped algorithms like KNN and MLP perform better. Also included for model compatibility were categorical variables, such as protocol kinds. The ensemble learning framework was able to effectively categorize a wide variety of attack types with little bias towards popular categories like SYN and benign traffic because of this preparation workflow, which made sure the dataset, was balanced, clean, and training-ready.

Figure 4 illustrates the distribution of flow time for both attacker and benign network traffic using logarithmic boxplots. The flow durations for attack traffic exhibit a very varied distribution, with flows ranging from ephemeral to exceedingly prolong. A multitude of elevated outliers and a comparatively extended median flow time for assaults signify ongoing attack activities, which is characteristic of distributed denial of service (DDoS), brute-force, or slow-rate attacks. The distribution of benign traffic is more concentrated and has a reduced interquartile range, indicating that the flow duration is more stable and foreseeable. In comparison to attack traffic, benign flows have fewer and less dispersed long-duration outliers.

The graph indicates that attack flows exhibit more variability and a broader range of durations, a crucial attribute for distinguishing malicious traffic from legitimate network activity.



(A) (B)

Figure 3: (A) Frequency of Label, (B) Percentage of Label

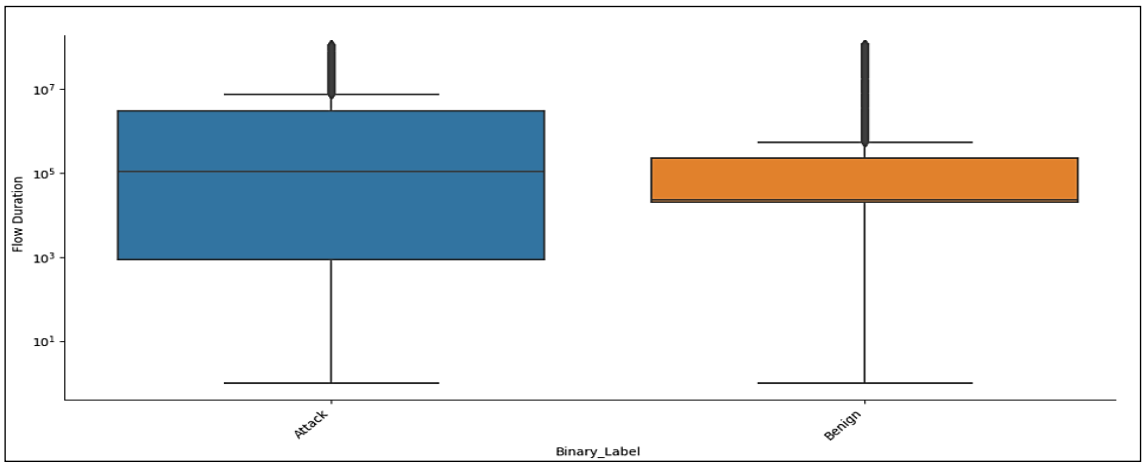


Figure 4: Flow Duration Distribution for Attack and Benign

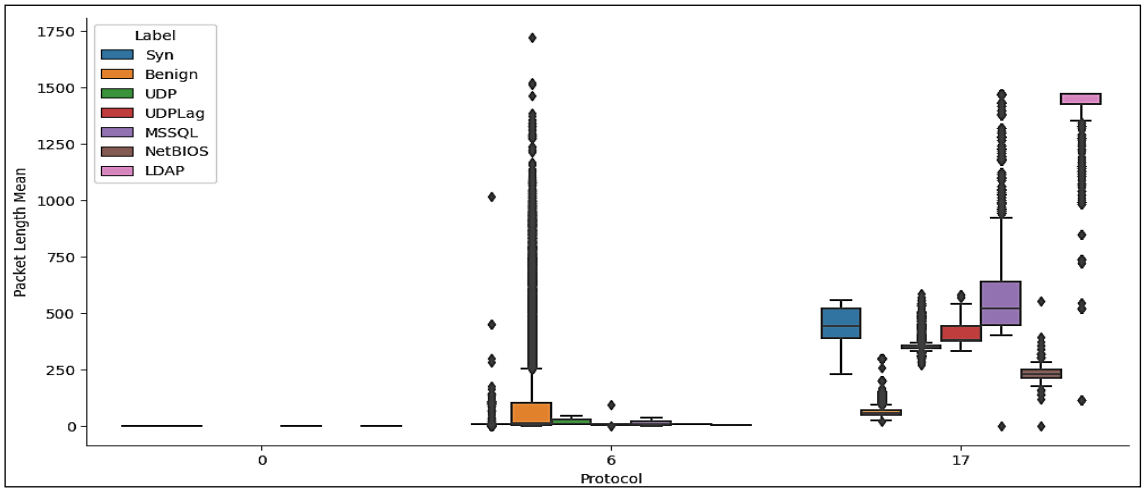


Figure 5: Packet Length Mean by Protocol and Attack Label

	Model	Accuracy	Precision	Recall	F1 Score	ROC AUC	CV Score
0	EML	0.992513	0.992720	0.992513	0.992388	0.991789	0.993625
1	LSTM	0.989903	0.990080	0.989903	0.989922	0.983676	0.990449
2	KNN	0.992128	0.992184	0.992128	0.992132	0.971321	0.993026
3	MLP Classifier	0.987636	0.987756	0.987636	0.987569	0.992681	0.988192

Figure 6: Performance Metrics

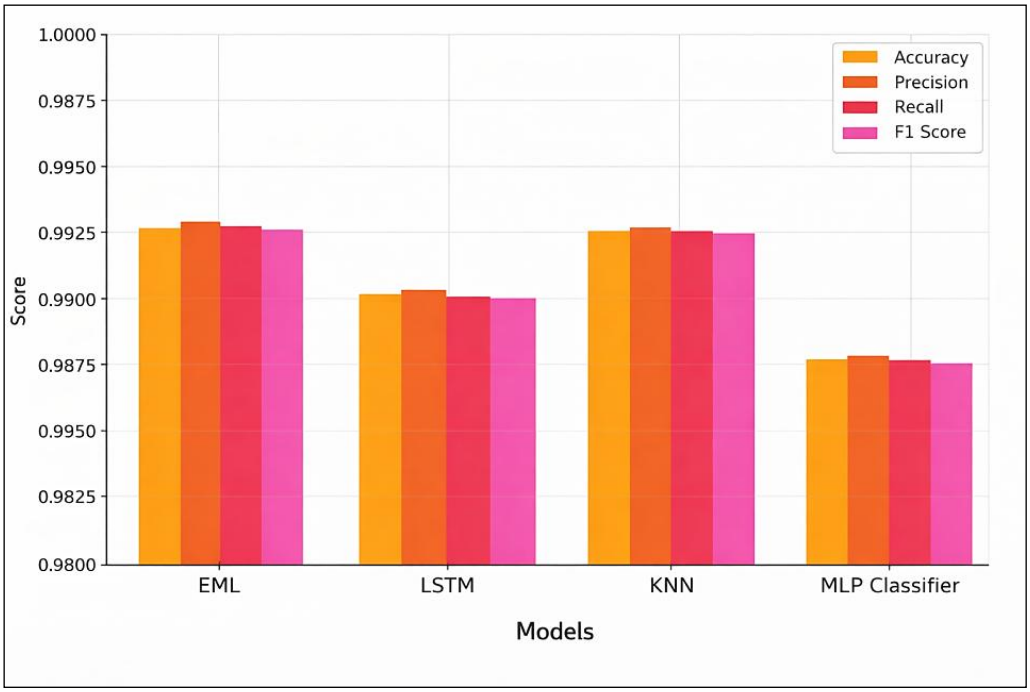


Figure 7: Performance Metrics

Figure 5 displays the distribution of mean packet lengths in the CIC-DDoS2019 dataset, broken down by attack categories and protocols. Although there is a great deal of variance in packet lengths between attack types, protocols 6 (TCP) and 17 (UDP) predominate in the sample. While SYN attacks (under TCP) have more concentrated but higher mean packet lengths, benign traffic under TCP exhibits a broad range of packet durations with numerous outliers. The mean packet length is more constant for UDP-based attacks, such as UDP flood, although there is notable variance for MSSQL and LDAP under UDP. Minimal change is seen in Protocol 0, despite its low use. This visualization shows how the ensemble classification model may use the mean of packet length to distinguish between various types of attacks and regular traffic.

Figures 6 and 7 present a comparative performance analysis of four models, ELM, LSTM, KNN, and MLP Classifier, based on several evaluation metrics for cyberattack detection using the CIC-DDoS2019 dataset. Among all models, EML consistently achieves the highest scores, with Accuracy (0.9925), Precision (0.9927), Recall (0.9925), and an impressive F1 Score (0.9923), demonstrating its ability to effectively combine the strengths of the individual classifiers. It also leads in ROC AUC (0.9917) and Cross Validation (CV) Score (0.9936), reflecting superior generalization and robust detection capability across different traffic scenarios.

KNN follows closely with strong performance (Accuracy: 0.9921, F1 Score: 0.9921), showing that its instance-based learning contributes well to detecting attack patterns. LSTM performs slightly lower (Accuracy: 0.9899), but its temporal sequence modeling gives it a solid edge in understanding time-dependent patterns, maintaining high recall and precision. MLP

Classifier scores slightly lower across all metrics (Accuracy: 0.9876, F1 Score: 0.9875), though it still performs competitively. Overall, the table highlights that while all models deliver strong results, the ensemble approach (EML) provides the most balanced and reliable performance, making it the optimal choice for strengthening cybersecurity and network defense.

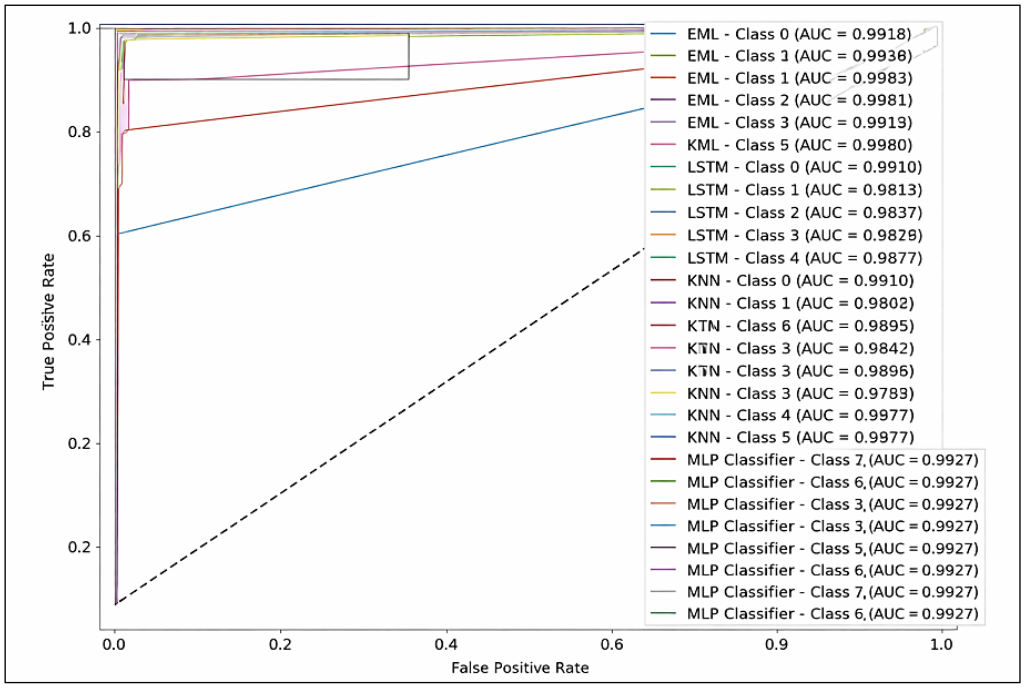


Figure 8: ROC Curve

Figure 8 illustrates the ROC (Receiver Operating Characteristic) curves for all models, ELM, LSTM, KNN, and MLP Classifier across multiple attack classes in the CIC-DDoS2019 dataset. The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR), showcasing each model's ability to distinguish between benign and malicious traffic across different DDoS attack types. The AUC (Area Under Curve) scores indicate overall model performance, with values close to 1.0 reflecting near-perfect classification.

The EML model dominates with a consistent AUC of 0.9918 across all classes, demonstrating superior and balanced detection for every attack type. MLP Classifier also performs strongly with an AUC of 0.9927, indicating its reliability in identifying complex patterns. LSTM achieves an AUC of 0.9837, leveraging its temporal sequence learning strength but showing slightly lower discrimination for some classes compared to EML. KNN, while still effective, scores 0.9713, reflecting solid but relatively weaker separation capability, especially for overlapping traffic patterns. The random guessing line (dashed diagonal) at 0.5 serves as a baseline, and all models significantly outperform it. Overall, the figure highlights that all models deliver excellent classification performance, but the ensemble model achieves the most consistent and reliable detection across all DDoS attack categories.

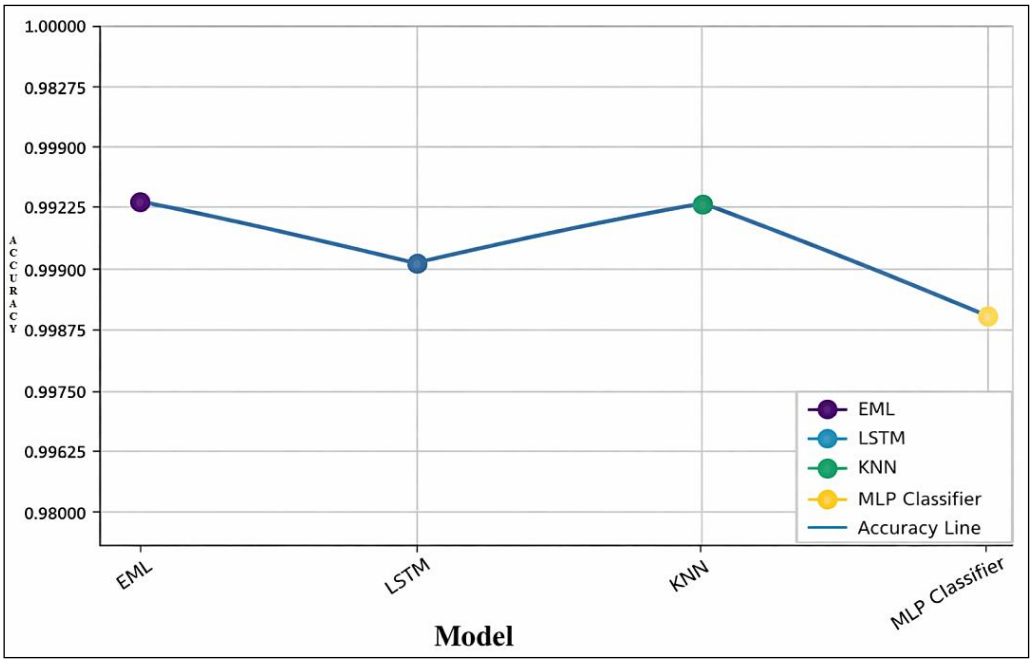


Figure 9: Accuracy Results

Figure 9 illustrates the accuracy scores of four models, EML, LSTM, KNN, and MLP Classifier, used in the ensemble learning framework for cybersecurity and network defense. The plot shows that the ELM achieves the highest accuracy at 0.9925, confirming its strength in combining multiple classifiers to deliver superior detection results. KNN follows closely with an accuracy of 0.9921, demonstrating strong performance due to its ability to classify based on distance metrics and neighborhood patterns. LSTM records a slightly lower accuracy of 0.9899, yet remains competitive, leveraging its capability to capture sequential dependencies in traffic data. The MLP Classifier scores the lowest accuracy at 0.9876, though it still performs well overall. The plotted accuracy line indicates only minor fluctuations among the models, showing that all four approaches are highly effective, but EML consistently outperforms the others, validating the benefits of combining KNN, LSTM, and MLP into a single, more powerful ensemble framework.

Limitations

a) Data processing in streaming and real-time. When processing continuous high-speed network streams, the LSTM model adds extra latency and memory expense even while it successfully captures temporal relationships.
b) The research recognizes that reaction time in real-world deployments, especially in ultra-low-latency situations, may be impacted by buffering and sequence-window selection.
c) The experimental evaluation is conducted using the CIC-DDoS2019 benchmark dataset, which, although comprehensive, may not fully capture the diversity and unpredictability of real-world network environments.
d) The study relies on offline training and testing, and therefore does not directly address challenges associated with real-time deployment, such as concept drift, latency constraints, and dynamic traffic behavior.

Conclusion

This study presents an ensemble machine learning architecture using the CIC-DDoS2019 dataset, combining KNN, LSTM, and MLP to enhance cybersecurity and network protection by accurately recognizing and classifying diverse DDoS attacks. The proposed method achieves superior performance across important evaluation metrics, such as accuracy, precision, recall, F1 score, and ROC AUC, by skillfully combining neighborhood-based pattern recognition, temporal traffic analysis, and nonlinear decision-making. It has a maximum detection accuracy of 99.25%. The experimental findings and visual assessments, such as ROC curves and comparative performance graphs, reveal that the ensemble model considerably reduces false positives and false negatives. This proves that the model can be used in real-world network security situations and

that it works well. The proposed architecture offers a dependable solution for deployment in modern cloud, IoT, and workplace networks where adaptive and precise intrusion detection is crucial. The study is limited by its reliance on an offline evaluation and benchmark dataset, which fails to adequately represent evolving attack strategies and real-time network dynamics. Future research will focus on real-time deployment, online and federated learning approaches, cross-dataset generalization, and computational optimization to make systems more scalable and adaptable to emerging cyberthreats.

Abbreviations

DDoS: Distributed Denial-of-Service, ELM: Ensemble Learning Model, EML: Ensemble Machine Learning, IDS: Intrusion Detection System, KNN: K-Nearest Neighbors, LSTM: Long Short-Term Memory, MLP: Multi-Layer Perceptron, NIDS: Network Intrusion Detection System, SOC: Security Operations Center, PCA: Principal Component Analysis.

Acknowledgement

The authors have no acknowledgements to declare.

Author Contributions

A Sivasangeetha: methodology, software implementation, D Joseph Pushparaj: methodology, software implementation, S Manjula: data curation, validation, visualization, Jasperline T: data curation, validation, R Saravanakumar: conceptualization, methodology, software implementation, data curation, validation, visualization, writing – original draft.

Conflict of Interest

The corresponding author declares that there is no conflict of interest on behalf of all authors.

Declaration of Artificial Intelligence (AI) Assistance

No generative AI or AI-assisted technologies were used in the preparation of this manuscript.

Ethics Approval

Not Applicable.

Funding

This research did not get any funding.

References

1. Singh J, Behal S. Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions. Computer Science Review. 2020 Jun 20;37:100279. https://doi.org/10.1016/j.cosrev.2020.100279
2. Hossain MdA, Sheikh MNA, Rahman SSM, Biswas S, Arman MdAI. Enhancing and measuring the performance in software-defined networking. International Journal of Computer Networks & Communications. 2018 Sep 30;10(5):27–40. https://doi.org/10.5121/ijcnc.2018.10502
3. Sheikh MNA, Hwang IS, Ganesan E, Kharga R. Performance Assessment for different SDN-Based Controllers. 2021 30th Wireless and Optical Communications Conference (WOCC). 2021 Oct 7;24–5. https://doi.org/10.1109/wocc53213.2021.9603050
4. Shahinzadeh H, Mahmoudi A, Asilian A, Sadrarhami H, Hemmati M, Saberi Y. Deep learning: a overview of theory and architectures. In2024 20th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP) 2024 Feb 21; p. 1-11. IEEE. https://doi.org/10.1109/AISP61396.2024.10475265
5. Dora VRS, Lakshmi VN. Optimal feature selection with CNN-feature learning for DDoS attack detection using meta-heuristic-based LSTM. International Journal of Intelligent Robotics and Applications. 2022 Jan 27;6(2):323–49. https://doi.org/10.1007/s41315-022-00224-4
6. Kuang C. Research on Network Traffic Anomaly Detection Method based on Deep learning. Journal of Physics Conference Series. 2021 Mar 1;1861(1):012007. https://doi.org/10.1088/1742-6596/1861/1/012007
7. Avcı İ, Koca M. Predicting DDOS attacks using machine learning algorithms in building management systems. Electronics. 2023 Oct 5;12(19):4142. https://doi.org/10.3390/electronics12194142
8. Admass WS, Munaye YY, Diro AA. Cyber security: State of the art, challenges and future directions. Cyber Security and Applications. 2023 Oct 6; 2:100031. https://doi.org/10.1016/j.csa.2023.100031
9. Abdullah M, Nawaz MM, Saleem B, Zahra M, Ashfaq EB, Muhammad Z. Evolution Cybercrime—Key Trends, Cybersecurity Threats, and Mitigation Strategies from Historical Data. Analytics. 2025 Sep 18;4(3):25. https://doi.org/10.3390/analytics4030025
10. Bagui S, Kalaimannan E, Bagui S, Nandi D, Pinto A. Using machine learning techniques to identify rare cyber-attacks on the UNSW-NB15 dataset. Security and Privacy. 2019 Oct 10;2(6). https://doi.org/10.1002/spy2.91
11. Homayoun S, Ahmadzadeh M, Hashemi S, Dehghantanha A, Khayami R. BoTShark: A deep learning approach for Botnet traffic detection. In: Advances in information security. 2018:137–53. https://doi.org/10.1007/978-3-319-73951-9_7

12. Peng W, Kong X, Peng G, Li X, Wang Z. Network Intrusion Detection Based on Deep Learning. International Conference on Communications, Information Systems and Computer Engineering (CISCE). 2019 Jul 1;431–5. https://doi.org/10.1109/cisce.2019.00102
13. Yang H, Wang F. Wireless network intrusion detection based on an improved convolutional neural network. IEEE Access. 2019 Jan 1;7:64366–74. https://doi.org/10.1109/access.2019.2917299
14. Rashid A, Siddique MJ, Ahmed SM. Machine and Deep Learning Based Comparative Analysis Using Hybrid Approaches for Intrusion Detection Systems. 3rd International Conference on Advancements in Computational Sciences (ICACS), Lahore, Pakistan. 2020 Feb 1;1–9. https://doi.org/10.1109/icacs47775.2020.9055946
15. Salih A, Zeebaree ST, Ameen S, Alkhyyat A, Shukur HM. A Survey on the Role of Artificial Intelligence, Machine Learning, and Deep Learning for Cybersecurity Attack Detection. 7th International Engineering Conference "Research & Innovation Amid Global Pandemic" (IEC), Erbil, Iraq. 2021 Feb 24;61–6. https://doi.org/10.1109/iec52205.2021.9476132
16. Gao X, Shan C, Hu C, Niu Z, Liu Z. An adaptive ensemble machine learning model for intrusion detection. IEEE Access. 2019 Jan 1;7:82512–21. https://doi.org/10.1109/access.2019.2923640
17. NSL-KDD, Datasets, Research. Canadian Institute for Cybersecurity. UNB. https://www.unb.ca/cic/datasets/nsl.html
18. Ahmad Z, Khan AS, Shiang CW, Abdullah J, Ahmad F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies. 2020 Oct 16;32(1). https://doi.org/10.1002/ett.4150
19. Kalinaki K, Thilakarathne NN, Mubarak HR, Malik OA, Abdullatif M. Cybersafe capabilities and utilities for smart cities. In: Advanced sciences and technologies for security applications. 2023:71–86. https://doi.org/10.1007/978-3-031-24946-4_6
20. Yost JR. The March of IDES: Early History of Intrusion-Detection Expert Systems. IEEE Annals of the History of Computing. 2015 Jul 13;38(4):42–54. https://doi.org/10.1109/mahc.2015.41
21. Rasheed A, San O, Kvamsdal T. Digital Twin: values, challenges and enablers from a modeling perspective. IEEE Access. 2020 Jan 1; 8:21980–2012. https://doi.org/10.1109/access.2020.2970143
22. Awad Z, Zakaria M, Hassan R. An enhanced ensemble defense framework for boosting adversarial robustness of intrusion detection systems. Scientific Reports. 2025 Apr 23;15(1):14177. https://doi.org/10.1038/s41598-025-94023-z
23. Ahmad Z, Khan AS, Shiang CW, Abdullah J, Ahmad F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies. 2020 Oct 16;32(1). https://doi.org/10.1002/ett.4150
24. Asmar M, Tuqan A. Integrating machine learning for sustaining cybersecurity in digital banks. Heliyon. 2024 Sep 1;10(17):e37571. https://doi.org/10.1016/j.heliyon.2024.e37571
25. Mahamkali N, Mudigonda KSP. An ensemble framework for effective detection and classification of cyber attacks using machine learning. In: Advances in computational intelligence and robotics book series. 2025. p. 65–96. https://doi.org/10.4018/979-8-3373-1807-3.ch004

How to Cite: Sivasangeetha A, Pushparaj DJ, Manjula S, Jasperline T, Saravanakumar R. Ensemble Machine Learning Approaches to Strengthen Cyber Security and Network Defense. Int Res J Multidiscip Scope. 2026; 7(1): 1499-1516. DOI: 10.47857/irjms.2026.v07i01.07401