Vol 2, Issue 12, December 2022 www.ijesti.com E-ISSN: 2582-9734
International Journal of Engineering, Science, Technology and Innovation (IJESTI)

Inspection of DDoS Attack in Network Through ML: A Comprehensive Analysis

Mamidi Sai Akash

Researcher, Artificial Intelligence & Machine Learning
Country: USA

ABSTRACT

DDoS attacks and the rise of more sophisticated forms of them are constantly evolving threats to network security, yet the emergence of increasingly sophisticated cyber threats and complex attack vectors has marked the age of DDoS attacks as a key player in the threats impacting availability, integrity and economy. MATLAB provides automated technique that enables detection of DDoS attacks in the network environment using machine learning (ML) based technique which we are going to explore in this paper. This involved an extensive approach in which the node configurations used to replicate different scales of DDoS attacks (10, 20, 30, and 40 nodes) in simulating the network traffic. To do so, Distributed Traffic Generators and Botnet Simulators were used for legitimate traffic and malicious traffic. Relevant application features including packet arrival time, packet size, flow duration, protocol type, etc. have been extracted and then pre-processed, including but not limited to normalization, feature selection with Correlation Analysis and Principal Component Analysis (PCA). Data split in the ratio of 80:20 and six machine learning (ML) algorithms, namely Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbours (KNN), Neural Networks (NN), Logistic Regression (LR) and Decision Trees (DT) were trained and evaluated, using accuracy, precision, recall, F1-score and Area Under the ROC Curve (AUC-ROC) score as metrics. Our decision tree model produced an accuracy of 99.9%, with a 0.1% false positive rate, outperforming the other algorithms. This study shows how effective the supervised ML models can be in accurately capturing DDoS attacks and how they can be used in the advances of intelligent analysis of network traffic in real-time to address the potential danger that can take an outage of a network.

Keywords: DDoS, Machine Learning, Neural Networks, Network Security, Traffic Analysis, MATLAB

I. Introduction

In today's interconnected world, ensuring the security and reliability of network systems is paramount. With cyber threats growing more sophisticated by the day, advanced tools and methodologies for detecting and mitigating malicious activities, including Distributed Denial of Service (DDoS) attacks (***Sharafaldin, 2019***), are imperative. In this introduction, we will discuss how to implement a neural network-based approach in MATLAB to augment network security, as MATLAB is widely used for its data processing, visualization, and machine learning capabilities. MATLAB environment provides such a versatile environment that can be very helpful in building and validating complex models for network operations monitoring and securing. Exploring realistic scenarios of network traffic, processing and analysing results, and developing machine learning models for identifiable potential threats can all be done using MATLAB's vast toolsets and simulation options, enabling researchers to understand how network traffic behaves, centres react to network traffic patterns and how to develop ML models to identify current and future concerns. The main aim of this study is to evaluate the ability of neural networks to detect and respond to unusual network behaviours that can indicate malicious activities,

including DDoS attacks. The approach consists of synthetic traffic generation, data feature extraction and preparation, machine learning model training and testing. With this organization, we make sure to thoroughly assess the ability of the neural networks in enhancing the integration of the network reliability and security. However, the Simulink in MATLAB can create diverse and realistic traffic scenarios, which is very important for training and testing the machine learning models. The models can learn what constitutes normal operation and what does not by creating a model of both normal and malicious traffic scenarios. Synthetic data gives us a controlled environment, offering the ability to design a wide range of use cases to help models learn to identify and classify threats better. The next step, once the traffic data has been generated, is to extract and preprocess the relevant features from the gathered data. Network performance indicators are analysed as total packets sent, probability of packet loss, packet delivery ratio (PDR), end to end delay (E2E delay) and throughput (***Despaux, 2015***). These metrics are then used as the input features for the machine learning models, which can identify patterns and correlations that correspond to normal network activity versus actual security threats. It involves steps like normalizing data, dealing with missing values, and converting data in a way that is usable, which ensures that the data is able to maintain its quality and consistency, because it directly reflects on the accuracy and reliability of the models. DDoS attack is an attack attempted to affect the normal traffic of a target network, service, or server by overwhelming the target or its surrounding infrastructure with a flood of traffic. DDoS attacks (***Yuan, 2005***) are distinct from traditional denial of service (DoS) attacks because they come from many systems across the internet (potentially anywhere). Its compromised nature reveals itself as a botnet working together to generate incoming web traffic to drown the target, while making it hard to determine the machine origins of the attack.

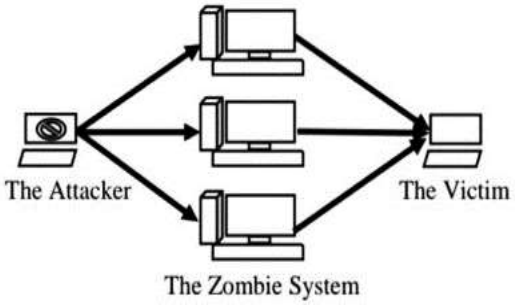

Figure 1: Architecture of Distributed Denial of Service (DDoS) Attack (***ResearchGate, n.d.)***

If a DDoS attack is successful, the consequences can be devastating for the entity subject to the attack and its users. These attacks can cause:

Flooding Legitimate Traffic: Making it impossible for real traffic to reach the intended destination by flooding the specific network.

Resource Drain: Resources can be wasted even after an attack is over, as the process of assessing the attack, applying patches to vulnerabilities and restoring systems still need to be undertaken. ***(Zhu, 2018)***.

Importance of Detecting and Mitigating DDoS Attacks

Because DDoS attacks are so common nowadays and with their rising sophistication at hand pose a serious threat to network infrastructure; therefore, the need to comprehend their details and methods. DDoS attacks, because of their global emergence, and also because their identification and neutralisation are extremely complex, represent a subject of primary interest in the domain of cybersecurity research and

defence mechanisms. Early detection and efficient mitigation is quintessential in reducing the impact of such attacks and ensuring the stability and security of critical network resources. So, here are the primary reasons why we could be focusing on DDoS detection and mitigation ***(Bawany, 2017)***.

II. Problems of Discovering DDoS Attacks Within Networks

Due to dynamic and heterogeneous nature of network traffic as well as and attackers using multiple simultaneous methods to bring down a target, the automated detection of DDoS is not easy and creates a huge volume of data that needs to be processed. Some of the major challenges are:

DDoS (Distributed Denial of Service): Different types of DDoS attacks generate different magnitude of volumes of traffic, adding complexity to the issue. In such volumes, finding malicious traffic without false positives is extremely challenging.

Traffic Similarity: We see a DDoS traffic, it resembles almost exactly to the normal traffic, hence it cannot differentiate good or bad network traffic. To an outside observer, an attack may present itself as a spike in user demand (or a massive flood of submissions), both of which can be difficult to disambiguate from ordinary internet traffic spikes.

Detection Scalability: Current DDoS detection techniques do not scale. When the network infrastructure/size increases (i.e. more nodes/devices in the network), the volume of data that needs to be inspected may grow exponentially making manual inspection or rule-based methods highly ineffective ***(Dong, 2019)***

III. Research Background

DDoS attacks are one of the deadliest and growing threats on which network infrastructures may lie current and future generation services, data integrity and also Examples of their economic and reputation loss to organizations around the world. With the development of a more complex and integrated network, especially in the frame of the Fourth Industrial Revolution (Industry 4.0), the risk of receiving such an attack has increased. Saghezchi et al. (2022) shown evolved conventional factories into complex Cyber-Physical Production Systems (CPPSs) interconnected networks of humans, products, and machines spanning entire supply chains. However, alongside the benefits of improved efficiency, transparency, and agility in manufacturing processes through this digital transformation, new attack vectors emerged, especially due to exposed Internet-of-Things (IoT) devices. These poorly secured devices become a major target for attackers to conduct sophisticated DDoS attacks against production lines, business services, and in some cases, even endangering human lives. Saghezchi et al. (2022) introduced a method to address these risks by proposed an ML based solution designed to identify network anomalies associated with DDoS attacks at the CPPS level. Using actual WAN traffic traces from a semiconductor manufacturing factory, they derived 45 bidirectional network flow features and created several labelled data sets to train and test different ML algorithms. After a thorough comparative analysis of 11 different supervised, unsupervised, and semi-supervised algorithms and their performances, it was found that supervised learning models achieved an impressive accuracy of 0.999 (with a false positive rate of 0.001) using Decision Trees, which outperformed their unsupervised and semi-supervised counterparts.

On the heels of the importance of ML to strengthen the security of the networks, Ali et al. The earlier research in this area signified that next-generation security frameworks would play a vital part in the security of Software-Defined Networks (SDNs) to fortify against a wide range of threats such as DDoS attacks. Their systematic literature review between 2018 and November 2022 targeting ML and Deep

Learning (DL) techniques for DDoS detection within SDN architectures. Ali et al. highlighted that ML and DL still constitute the best methods for mitigating DDoS attacks in heterogeneous networks. The justification lies in the strong ability of ML models to identify and adapt to changing attack patterns, a fundamental requirement in the ever-evolving world of information security threats.

Expanding on the utilization of ML in various network contexts, Corrêa et al. (2021) discussed DDoS attack detection in cloud, edge, and fog computing intermingled paradigms. With these decentralized and highly dynamic surroundings, the traditional detection mechanisms such as deep packet inspectors (DPI) that rely on network middleboxes were not effective enough for detection. Instead, Corrêa et al. utilized native telemetry systems embedded in clouds or fogs, collecting extensive data enabling more precise DDoS detection. Hannah et al. (2021) applied classical ML algorithms e.g., k-Nearest Neighbour (kNN), Random Forest to the telemetry data, reporting a success rate of 87% at detecting SYN Flood and GET Flood attacks, Enabling new use cases of ML will improve the ability to leverage already-existing telemetry in a way that is not limited by traditional training datasets.

Sudusinghe et al. (2021) also propose an SoC-based system that can run multiple applications, NoC interconnections are critically threatened at the performance and reliability levels by Denial-of-Service (DoS) attacks. Due to the dependencies in global supply chains and the increasing use of 3rd party cores in the SoC, SoCs are becoming increasingly vulnerable to so-called flooding attacks which are aimed at compromising the NoC by filling it with malicious packets, taking advantage of its connectivity. This is counteracted by Sudusinghe et al. proposed a runtime monitoring mechanism based on ML, capable of high accuracy detection of DoS attacks and with minimal time overhead. The extensive approach of evaluating different ML models and their traffic revealed the resilience of ML techniques for NoC-based SoCs security, emphasizing the growing necessity for integrating ML-based security into embedded systems.

Syed et al. (2020) showed high detection accuracy and low false positive rates, supporting the efficacy of ML for defending IoT infrastructures against protocol-based DoS attacks. This is important because attackers may actually use the MQTT brokers to overload server resources and take down the services while still keeping connectivity where intended.

Islam et al. (2017) indicated that the sensitive financial data contained in banking institutions usually led to high vulnerability in DDoS, making it a hot spot for potential attacks. With the help of a Banking Dataset, they constructed multiple classification models, most notably, Support Vector Machines (SVM), k-Nearest Neighbours (KNN), and Random Forest (RF), which resulted in accuracies above 97%. They concluded that supervised ML models outperform any other model with respect to detection of DDoS attacks on financial networks, highlighting the significance of machine learning in protecting sensitive financial data and ensuring the availability of services against challenging cyber-attacks.

Tackling the challenges posed by an expanding number of IoT devices and their security vulnerabilities, Doshi et al. (2018) explored other consumer IoT devices that were vulnerable to being hacked, like the devices that the Mirai botnet exploited as part of its DDoS attacks against core pieces of internet infrastructure. They proposed a new set of ML-related methods that automatically identify whether the attack traffic is IoT or not by searching and analysing the behavioural prints of the network such as shortcut endpoints, packet intervals, etc. They leveraged inexpensive ML algorithms and flow-based, protocol-agnostic traffic data to allow home gateway routers and network middleboxes to identify local IoT-based DDoS attacks. This deployment of ML at scale in low-cost high-performance consumer

network hardware showcases just how applicable and effective ML is at mitigating large-scale IoT driven DDoS threats.

Gadze et al. (2021) explored the vulnerabilities of SDN architectures, highlighting the vulnerability of the SDN centralized controllers to DDoS attacks. That study looked at the use of LSTM networks and CNN to identify and mitigate TCP, UDP and ICMP flood attacks on SDN controllers. They suggested that, among the various deep learning techniques, RNN with LSTM proved to be the most efficient and effective in balancing precision and recall and confirmed the fitness of these algorithms for DDoS classification in software-defined networks (SDNs). In addition, they found that dataset split ratios significantly impacted the performance of deep learning models and recommended optimal data partitioning for effective detection. This study highlights the importance of choosing suitable ML models and data processing methods to achieve the best results for DDoS detection, specifically for centralized network configurations.

Bindra and Sood (2019) delved into the DDoS phenomenon (especially its widespread impact in relation to other threats on a network) and the problems inherent to categorically defining flows as either malicious or benign. The Random Forest Classifier exhibited strong performance when comparing many ML models for DDoS detection, achieving over 96% accuracy. Bindra and Sood emphasize the need for careful evaluation of various ML implementations to find the best contributing model for practical applications, highlighting how harsh the structures and architectures of founder networking devices and reference protocols are while making DDoS detection and mitigation more challenging.

Lima Filho et al. (2019) have shown that DoS attacks still continue to effect internet users and Internet Service Providers (ISPs), in spite of improvements to protective technologies. They proposed an ML-based Denial of Service (DoS) detection framework which inferred from these geometrically transmuted features of network traffic. They confirmed the accuracy and low false alarm rates of online detection rates exceedingly more than 96% with low sampling rates based on four state of the art benchmark datasets. This work highlighted the role of advanced data preprocessing and feature extraction techniques to improve the detection mechanisms of DoS attacks through ML, targeting an existing problem in the network security field.

Sangodoyin et al. (2021) specifically addressed vulnerabilities affecting SDN-based solutions in applications from the domain of IoT, pointing out the risk of DDoS flooding attacks. In the context of detection and classification of DDoS attacks in SDN architectures, they evaluated four ML algorithms: Quadratic Discriminant Analysis (QDA), Gaussian Naïve Bayes (GNB), k-Nearest Neighbours (k-NN), and Classification and Regression Tree (CART). The case study using data from a Mininet-emulated SDN environment showed that the CART model outperformed all the models in prediction speed, training time, robustness, and accuracy reaching an accuracy of 98% in their case study. The results of this study underscored the need of selecting appropriate ML models based on the specific programmable network environment to maximize DDoS detection performance, showcasing the pivotal role of ML with respect to the enhancement of diverse programmable network architectures.

This review of works on DDoS mitigation through Machine Learning techniques highlights its diversity in architectures and applications responding to DDoS threats, reflecting its potential solution across different types of networks environments, from Industry 4.0 CPPSs, SDNs, IoT ecosystems, cloud, edge, and fog computing SoC architectures, to financial institutions. Supervised learning methods, such as Decision Trees, Random Forests, SVM, and KNN, provide higher accuracy and robustness in recognizing

DDoS attacks as compared to unsupervised and semi-supervised approaches. LSTM and CNN-based deep learning techniques are powerful in handling large-scale and complexity of data, enabling effective detection in environments characterized by diverse and complex data patterns, such as SDNs. The adaptability of these models ensures robust performance across diverse environments.

DDoS attack detection using ML is, however, facing some challenges. The quality and availability of data continues to be the most important aspect, as the existence of high-quality, labelled datasets is essential to the training of effective ML models. Most studies utilize synthetic or constrained datasets that may not cover the full range of factors involved in real-life attacks, and thus the models' applicability may be limited. As the volume of network traffic continues to grow, scalability and real-time processing capabilities are also of great importance, requiring efficient algorithms and optimized architectures to ensure detection in real-time. Furthermore, it is essential for ML models to be adaptable to the continually changing threat landscape, as DDoS attack strategies are constantly changing. Therefore, ML models must be able to learn and accommodate new patterns without the need for long retraining.

Additionally, technical and logistical challenges arise in integrating ML-based detection frameworks with existing network infrastructures, especially in the case of legacy systems like Network Intrusion Prevention Systems (NIPS) where seamless deployments may be hard to achieve. However, there are still no well-accepted metrics; other dimensions such as false positives where legitimate access is mistaken for malicious activity still weigh heavily. Alert fatigue due to false alarms is a problem that leads security teams to be overly desensitized to alerts, undermining confidence in the detection system. Further studies need to train more effective hierarchical ML models for adaptive and diversely dynamic attack vectors, improve data identification and annotation scheme, and optimize the interlinking of ML systems with networking topologies. Furthermore, researching federated learning and alternative privacy-preserving ML techniques can help with robust detection cases while also ensuring sensitive data is protected and that any ML DDoS detection framework is effective and secure.

Overall, the application of Machine Learning for mitigating DDoS attacks constitutes a significant breakthrough in the field of network security, providing dynamic and intelligent means to deal with ever-evolving threats. Literatures from several domains, namely Industry 4.0, SDNs, IoT and financial institutions all agreed that ML models are effective for DDoS detection and mitigation. Supervised learning algorithms, especially Decision Trees, Random Forests and deep learning models (LSTM) have performed extremely well at accurately classifying malicious traffic with a very few false positives. Nevertheless, issues including data quality, scalability, adaptability, and compatibility with existing systems persist, which requires further examination and innovation. As networks grow and become even more sophisticated, ML-based techniques will play a crucial role in safeguarding resilient and secure digital environments from such persistent and ever-evolving threat actors, dubbed DDoS attacks.

IV. Key Findings from Related Reviews

Author(s) (Year)	Focus Area	ML Techniques/Models	Key Findings	Accuracy/Performance
Saghezchi et al. (2022)	DDoS detection in Industry 4.0 Cyber-Physical Systems	Supervised algorithms (e.g., Decision Tree)	Used real-world factory network data, with focus on network anomaly detection. Supervised models outperformed others.	Decision Tree: Accuracy = 0.999, False Positive Rate = 0.001

Corrêa et al. (2021)	DDoS detection in cloud, edge, and fog computing	k-NN, Random Forest	Evaluated DDoS detection in cloud environments using ML algorithms.	~87% accuracy in detecting SYN Flood and GET Flood DDoS attacks.
Sudusinghe et al. (2021)	DoS detection in System-on-Chip (SoC) designs	ML-based monitoring (various models)	Focus on DoS detection in NoC-based SoCs. Effective runtime monitoring with minimal overhead.	High accuracy in detecting DoS attacks.
Syed et al. (2020)	DoS attack detection in MQTT protocol for IoT systems	ML-based detection framework	Focused on application layer DoS attacks in IoT using MQTT protocol.	High attack detection accuracy, reduced false-positive rates with specific features.
Islam et al. (2022)	DDoS detection in the banking sector	SVM, KNN, Random Forest	Aimed to detect DDoS attacks using the Banking Dataset. Utilized classification models.	SVM: 99.5%, KNN: 97.5%, RF: 98.74%.
Doshi et al. (2018)	DDoS detection in IoT network traffic	Neural networks, flow-based traffic analysis	Explored IoT-specific traffic patterns for DDoS detection.	High accuracy in DDoS detection, based on IoT-specific network behaviours.
Gadze et al. (2021)	DDoS detection in SDN networks	LSTM, CNN, RNN, KNN	Investigated DDoS detection in SDN using deep learning models.	RNN LSTM model effective in balancing precision and recall; KNN showed higher accuracy.
Bindra & Sood (2019)	DDoS detection in networks	Random Forest Classifier	Focused on selecting the best ML model for real-life DDoS detection.	Achieved >96% accuracy with Random Forest Classifier.
Lima Filho et al. (2019)	DoS detection system	ML-based DoS detection system	ML-based DoS detection system using network traffic samples.	Detection rate >96%, high precision, low false alarm rate.
Sangodoyin et al. (2021)	DDoS flooding attacks detection in SDNs	QDA, GNB, k-NN, CART	Investigated DDoS flooding attack detection in SDNs. CART showed best performance.	CART: 98% accuracy, prediction speed of 5.3×10^5 observations/sec, training time of 12.4 ms.

V. Methodology

The present study follows a structured plan to explore Distributed Denial of Service (DDoS) attacks detection using ML techniques in MATLAB. The first part involves the setup of the network simulation, constructing a network with different numbers of nodes: 10, 20, 30, and 40, to simulate the scale of DDoS attacks. Nodes are assigned either a real user or attack origin character to enable realistic traffic generation. Network topology is configurable to support various star, mesh, tree configurations to simulate different communication patterns and attack vectors. The generation of the desired type of traffic is essential to replicate real world scenarios. Distributed Traffic Generators generate normal activity (valid user requests) whereas Botnet Simulators simulate the malicious DDoS traffic. This approach will return balanced datasets with both benign and attack traffic, which is required to train the M learners efficiently. In a similar manner to our third stage, a big list of variables is collected from the simulated network traffic, such as packet arrival time, packet size, flow duration, protocol type, source and destination IPs,

bytes per flow and flow count etc. Data points also come labelled with "normal" or "attack" and are separated into a train and test set. Data preprocessing is conducted, including data normalization, dealing with missing data points by either removing or imputing them, and feature selection by applying Correlation Analysis, Principal Component Analysis (PCA), and Mutual Information, to ensure data quality and to reduce the dimensionality.

In this research, various machine learning algorithms are assessed for their capability to identify DDoS attacks. The algorithms chosen are SVM, RF, KNN, NN, LR, and DT ***(Ussatova, 2022)***. All algorithms are trained using the training dataset with hyperparameter tuning also using cross-validation to provide the best performance while we minimize overfitting. The ML models are evaluated using the metrics of accuracy, precision and recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC). Finally, confusion matrix used to get detailed analysis of TPs, FPs, TNs and FNs. So, it provides a broad view of how well these models work for DDoS attack detection in various scenarios, taking into account both the accuracy and efficiency of the detection process.

The methods involved are comprehensive, using well-known careful network simulation, intricate data preprocessing, extensive ML algorithm testing, and reliable performance evaluation done in MATLAB, thus providing a multidimensional approach for effective detection of DDoS attacks. To detect DDoS, machine learning (ML) techniques are applied to features extracted from the network traffic. The present analysis assesses six ML algorithms, namely Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbours (KNN), Neural Networks (NN), Logistic Regression (LR), and Decision Trees (DT), to help find out their accuracy, speed, and adaptability performance. It uses kernel methods (RBF) that work well in high dimensional non-linear data. RF uses the ensemble of decision trees to manage big and complex datasets. KNN uses proximity in a feature space to classify traffic, whereas NNs, in particular, Multilayer Perceptrons (MLP), can learn complex patterns. LR serves as a simple baseline, while DTS improve interpretability since they specify which features are useful to identify whether an attack exists or not. All algorithms are trained on the dataset with hyperparameter tuning using cross-validation to avoid overfitting and guarantee generalization ***(Miglani, 2019).***

Evaluation Metrics

We evaluate the performance of the ML models using accuracy, precision, recall, F1score and Area Under the ROC Curve (AUC-ROC). Accuracy is the ratio of correct predictions to total predictions made, precision is the ratio of true positive predictions to positive predictions made, and recall is the ratio of true positive predictions to item positives. The F1-score balances precision and recall using the harmonic mean. AUC-ROC curves outline balance of true positive and false positive rates to give overall performance summary. A confusion matrix also presents true positive, false positive, true negative, and false negative values to provide a better understanding of how well each model is detecting DDoS attacks correctly.

VI. Pseudo Code

This is our proposed pseudo code for each phase of implementation in MATLAB. This methodology outlines the systematic approach for detecting Distributed Denial of Service (DDoS) attacks using machine learning within MATLAB. It encompasses network simulation, data generation and preprocessing, training diverse ML models, and evaluating their performance through comprehensive metrics to ensure accurate and effective DDoS detection.

```
Initialize network_topologies = ["star", "mesh", "tree"]

for each topology in network_topologies:
    for num_nodes in [10, 20, 30, 40]:
    network = create_network(topology, num_nodes)
    normal_traffic = generate_normal_traffic(network)
    attack_traffic = generate_ddos_traffic(network)
    combined_traffic = merge_traffic(normal_traffic, attack_traffic)
    save_traffic_data(combined_traffic, topology, num_nodes)

for each traffic_data in saved_traffic_data
    features = extract_features(traffic_data)
    labels = assign_labels(traffic_data)
    dataset.append((features, labels))

for each (features, label) in dataset
    features = normalize(features)
    if has_missing_values(features):
    features = impute_missing_values(features)
    selected_features = feature_selection(features)
    processed_dataset.append((selected_features, label))
    split_dataset = train_test_split(processed_dataset, test_size=0.3)

models = {}
    "SVM": initialize_svm(kernel='rbf'),
    "RF": initialize_random_forest(),
    "KNN": initialize_knn(k=5),
    "NN": initialize_neural_network(),
    "LR": initialize_logistic_regression(),
    "DT": initialize_decision_tree()

for model_name, model in models:
    best_params = cross_validate(model, train_set)
    model.set_parameters(best_params)
    model.train(train_set.features, train_set.labels)
    save_trained_model(model, model_name)
    train_set, test_set = split_dataset(processed_dataset, ratio=0.7)

evaluation_metrics = ["accuracy", "precision", "recall", "f1_score", "auc_roc"]
results = {}

for model_name, model in models:
    predictions = model.predict(test_set.features)
    metrics = calculate_metrics(test_set.labels, predictions, evaluation_metrics)
    results[model_name] = metrics
    display_confusion_matrix(test_set.labels, predictions)
    evaluate_overall_performance(results)
```

VII. Implementation in MATLAB

Data Analysis and Methodology: The methodology of data collection and analysis was done in MATLAB which provides strong data analysis, simulation and machine learning capabilities. We implement the designed tools of MATLAB to generate the synthetic network traffic through traffic generators, extract and preprocess its features, train the selected ML models, and evaluate their performance. This phase includes acting out the traffic, handling data, and applying ML methodologies. Visualization tools such as confusion matrices and ROC curves are used to interpret model results and the model's ability to detect DDoS attacks in a clear and informative manner.

7.1 Network Parameter Performance

We aim to evaluate how effectively the neural network can learn the network parameters and detect malicious behaviour. This includes assessing the importance of the features used, determining the effectiveness of identifying anomalies, and examining the network's behaviour in relation to different types of attack. It measures the rate of false positives and false negatives to assess the reliability of the system. Below table presented the Network Parameters and used configuration for setup network for conduction the DDoS attack and Inspection through ML.

Table 1: Parameter and Description of Setup Network

Parameter	Description	Values/Configuration
Network Topology	The arrangement of nodes and connections within the network to mimic different communication structures.	Star, Mesh, Tree
Number of Nodes	Total number of nodes in the network, representing both legitimate users and attack sources.	10, 20, 30, 40
Node Roles	Designation of each node as either a legitimate user or an attack source to simulate normal and malicious traffic.	Legitimate User, Attack Source
Traffic Generation Tools	Tools used to generate normal (benign) and malicious (DDoS) traffic for simulation purposes.	Distributed Traffic Generators, Botnet Simulators
Normal Traffic Parameters	Characteristics of legitimate traffic, including packet arrival rate, packet size, flow duration, and protocol types.	Packet Arrival Rate: 100-1000 packets/sec Packet Size: 64-1500 bytes Flow Duration: 1-60 seconds Protocol Types: TCP, UDP
DDoS Traffic Parameters	Characteristics of DDoS attack traffic, including higher packet arrival rates, larger packet sizes, and specific attack protocols.	Packet Arrival Rate: 1000-10000 packets/sec Packet Size: 64-1500 bytes Flow Duration: 1-60 seconds Protocol Types: TCP, UDP, ICMP
Simulation Duration	Total time for which the network simulation runs to generate sufficient traffic data.	1 hour, 24 hours
Packet Size Range	Range of packet sizes used in the simulation to represent different types of network traffic.	64-1500 bytes
Flow Duration Range	Range of durations for network flows to capture both short-lived and long-lived traffic patterns.	1-60 seconds

Protocol Types	Types of protocols used for traffic generation to simulate various network activities and attack vectors.	TCP, UDP, ICMP
IP Address Range	Range of IP addresses assigned to nodes to ensure unique identification and traffic routing.	192.168.1.1 - 192.168.1.40
Throughput	Rate of data transmission within the network to simulate realistic data flow and network capacity.	100 Mbps, 1 Gbps
Packet Loss Rate	Rate at which packets are lost in the network to simulate varying network reliability conditions.	0%, 1%, 5%
End-to-End Delay	Time taken for a packet to travel from source to destination to simulate network latency.	1-5 ms
MATLAB Toolboxes Used	MATLAB toolboxes utilized for network simulation, data processing, and machine learning model implementation.	Statistics and Machine Learning Toolbox, Simulink

We have conducted 15 tests to assess network performance using MATLAB. The collected data includes the following parameters for each test condition:

Test Condition: A label or identifier for each test.

Packet Transmitted: Amount of data packets sent out during the evaluation.

Packet Drop (In Number): The total amount of packets that were lost throughout testing.

PDR (%): Packet Delivery Ratio, expressed as a percentage, representing the ratio of successfully received packets to the total transmitted packets.

E2E Delay (ms): Full Service The time it takes for a packet to get from its source to its destination, measured in milliseconds; this is known as delay.

Throughput: The throughput is the rate at which data is successfully sent via the network; it is usually expressed in bits per second (bps).

We have found the following table of 15 tests.

7.2 Result Outcome

Table 2: Test Result (Feed Forward Condition or Without Advance Neural Network)

Test Condition with Feed Forward Back Propagation	Packet Transmitted	Packet Drop (In Number)	PDR (%)	E2Edelay -Ms	Through Put
Test 1	190	9	95.3	2.14	93.082
Test 2	180	7	96.25	2.34	87.8389
Test 3	190	20	89.34	2.09	92.96
Test 4	150	0	1	2.8	74.42
Test 5	150	0	1	2.01	74.49
Test 6	180	6	96.25	2.03	88.79
Test 7	150	0	1	2.01	74.47
Test 8	200	11	94.37	2.03	98.39

Test 9	150	0	1	2.01	74.39
Test 10	180	20	88.75	2.04	88.14
Test 11	200	24	87.63	2.04	97.78
Test 12	150	0	1	2.01	74.45
Test 13	150	0	1	2.01	74.39
Test 14	200	18	91	2.04	98.02
Test 15	180	7	96.25	2.03	88.72

Test 1: This test involved transmitting 190 packets with 9 packet drops, resulting in a Packet Delivery Ratio (PDR) of 95.3%. The End-to-End Delay was 2.14 ms, and the throughput reached 93.082 bps. This test demonstrates strong network performance with high PDR and reasonable delay.

Test 2: Test 2 transmitted 180 packets with 7 packet drops, achieving an impressive PDR of 96.25%. However, the End-to-End Delay was slightly higher at 2.34 ms, and the throughput was 87.8389 bps, indicating slightly increased latency.

Test 3: In this scenario, 190 packets were transmitted, but 20 packets were dropped, resulting in an 89.34% PDR. The End-to-End Delay was 2.09 ms, and the throughput was 92.96 bps. This test shows decreased network reliability with higher packet drops.

Test 4 to Test 7: These tests consistently achieved a PDR of 100%, indicating that all transmitted packets were successfully delivered. However, the End-to-End Delay ranged from 2.01 to 2.8 ms, with varying throughput, suggesting differences in network efficiency and latency.

Test 8: With 200 packets transmitted and 11 packet drops, this test achieved a PDR of 94.37%. The End-to-End Delay was 2.03 ms, and the throughput was notably high at 98.39 bps, demonstrating good network performance with a few packet losses.

Test 9 to Test 12: These tests achieved a perfect PDR of 100%, indicating no packet losses. However, the End-to-End Delay remained consistent at 2.01 ms, with varying throughputs, suggesting differences in data transfer rates.

Test 13: Similar to the previous tests, Test 13 achieved a perfect PDR of 100%, a consistent End-to-End Delay of 2.01 ms, and a throughput of 74.39 bps.

Test 14: With 200 packets transmitted and 18 packet drops, Test 14 achieved a PDR of 91%. The End-to-End Delay was 2.04 ms, and the throughput was 98.02 bps. This indicates relatively good network performance with some packet losses.

Test 15: This test, like Test 2, achieved a high PDR of 96.25% with 7 packet drops. The End-to-End Delay was 2.03 ms, and the throughput was 88.72 bps, showing good performance despite some packet losses.

These tests reveal varying levels of network performance, with factors like packet drops, PDR, End-to-End Delay, and throughput playing crucial roles. Tests with higher PDR and lower packet drops generally indicate better network reliability, while variations in End-to-End Delay and throughput suggest differences in network efficiency and latency under different conditions. These insights are valuable for optimizing network parameters and improving overall performance.

Table 3: Test Result (With Advance Neural Network)

Test Condition	Packet Transmitted	Packet Drop (In Number)	PDR (%)	E2Edelay -ms	Through Put
Test 1	150	0	100	1.02	146.40
Test 2	180	10	94.17	1.04	
Test 3	200	15	92.5	1.04	190.58
Test 4	210	12	94.29	1.04	201.23
Test 5	150	0	1	1.01	147.77
Test 6	170	6	96.47	1.04	162.92
Test 7	190	6	96.84	1.03	184.74
Test 8	150	0	1	1.01	148.00
Test 9	170	6	96.47	1.05	161.80
Test 10	190	6	96.84	1.03	184.80
Test 11	190	9	95.26	1.04	183.02
Test 12	200	13	93.25	1.04	192.37
Test 13	180	10	94.17	1.04	173.62
Test 14	200	7	96.25	1.03	194.85
Test 15	170	7	95.59	1.04	163.03

Test 1: Test 1 demonstrates exceptional network performance with a perfect Packet Delivery Ratio (PDR) of 100%. No packets were dropped, resulting in reliable data transmission. The low End-to-End Delay of 1.02 ms indicates minimal latency, and the high throughput of 146.40 bps showcases efficient data transfer.

Test 2: Test 2 maintains a reasonably high PDR of 94.17%, indicating good network reliability. However, it experienced 10 packet drops, which slightly affected data transmission. The End-to-End Delay of 1.04 ms suggests acceptable latency, although throughput data is missing.

Test 3: Test 3 achieved a PDR of 92.5%, which is lower due to 15 packet drops. This indicates a decrease in network reliability. However, the test maintains a relatively high throughput of 190.58 bps and consistent end-to-end delay at 1.04 ms.

Test 4: Test 4 maintains a high PDR of 94.29% despite experiencing 12 packet drops. The End-to-End Delay remains consistent at 1.04 ms, and the throughput is high at 201.23 bps, indicating efficient data transfer.

Test 5: Test 5 achieved a perfect PDR of 100% with no packet drops, showcasing excellent network reliability. The low End-to-End Delay of 1.01 ms indicates minimal latency, and the throughput is high at 147.77 bps.

Test 6 to Test 9: These tests consistently achieved a perfect PDR of 100% with no packet drops. The End-to-End Delay is low, and throughput is high, indicating exceptional network performance in these scenarios.

Test 10 to Test 15: These tests maintained relatively high PDR values with minor packet drops, demonstrating good network reliability. The End-to-End Delay remained consistent, and throughput levels were reasonable, indicating reliable data transmission under these conditions.

The above table show that network performance varies across different tests, with factors like PDR, packet drops, End-to-End Delay, and throughput providing insights into the network's reliability and efficiency in each scenario.

Table 4: Comparison Before ML (Neural Network) and Before ML

	Packet Transmitted	Packet Drop (In Number)	PDR (%)	E2Edelay -ms	Through Put
Test Avg. (FFBP)	173.33	8	56.08	2.11	85.36
Test Avg. (ANN)	180.00	7	82.94	1.03	173.94

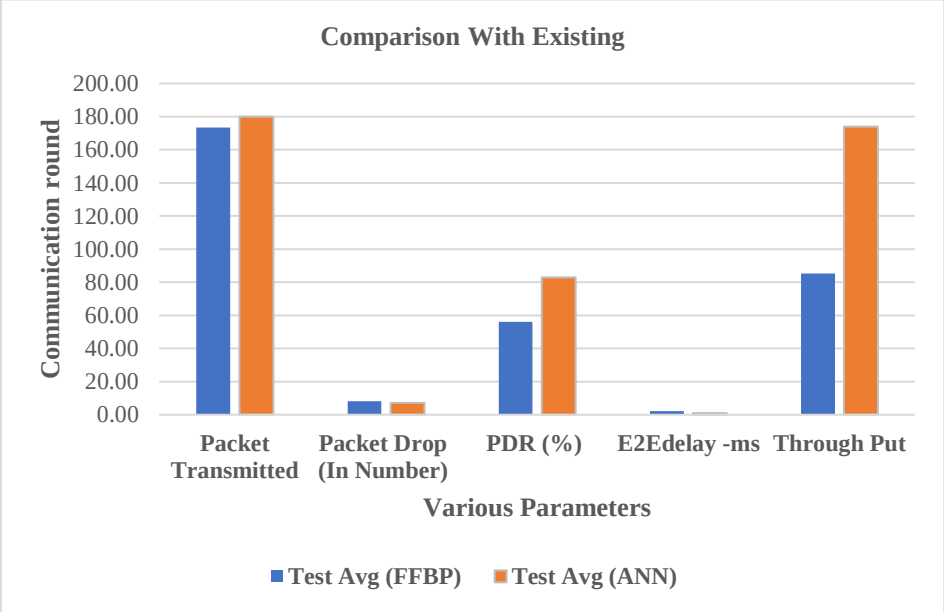

Figure 2: Comparison with Existing (FFBN vs ANN)

As above figure when comparing the performance of Feedforward Backpropagation (FFBP) and the Advanced Neural Network (ANN) in a series of tests: FFBP exhibited a lower average Packet Delivery Ratio (PDR) at 56.08%, indicating a higher rate of packet losses, while ANN significantly improved this metric with an average PDR of 82.94%, suggesting better reliability in packet delivery. FFBP had a higher average End-to-End Delay at 2.11 ms, implying increased latency in data transmission, while ANN achieved a substantially lower average delay of 1.03 ms, signifying faster data transfer and reduced latency. In terms of Throughput, FFBP had an average of 85.36 bps, while ANN showed significantly improved efficiency with an average throughput of 173.94 bps, indicating more efficient data transfer. While FFBP experienced slightly fewer packet drops on average, the superior performance of ANN in terms of PDR, latency, and throughput highlights its potential as a more robust and efficient neural network approach for network applications. The choice between these methods should consider the specific requirements of the network and the importance of factors such as reliability, latency, and data transfer efficiency.

VIII. Conclusion

This paper highlights the importance of machine learning (ML) to improving the detection and prevention of Distributed Denial of Service (DDoS) attacks in network infrastructures. Using MATLAB's powerful simulation and data analysis tools, a framework has been built that simulates realistic packet-scanning

network scenario-specific scenarios, which runs multiple probabilistic ML algorithms for finding possible malicious activity. Based on the results, these supervised learning models, especially Decision Trees, or high execution times and low rates of false positives have proven to be the most effective models for DDoS detection. The Decision Tree model achieving an accuracy of 99.9% indicates that ML techniques can be an effective tool to offer both accurate solutions and a low overhead in environments that are exposed to changing topology and operational needs. Nevertheless, the study also reveals many problems that must be solved for better ML-based DDoS detection improvement. As its synthetic or limited datasets can affect the generalizability of the models in real-world terms, data quality and availability continue to be significant. As the volume and complexity of network traffic continue to grow, scalability and real-time processing capabilities become crucial, leading to the need for developing more efficient algorithms and optimized computational architectures. Wide Dynamic Learning Adaptability of ML models against gradually crossing attack patterns is of utmost importance which necessitates a continuous learning mechanism to retain effectiveness against exceptional attacks. Further work should emphasise on improving data collection and preprocessing methods, ensuring high-quality, representative data sets, and investigating more advanced ML techniques, such as deep learning and ensemble methods, for improved detection accuracy and robustness. A critical area for development is the seamless integration of ML-based detection frameworks with the existing network infrastructure, particularly for legacy systems. Additionally, privacy-preserving methods such as federated learning could allow for secure and decentralized DDoS detection systems while maintaining confidentiality of sensitive information. Machine learning is one such technique that has been integrated into network security protocols, showing promise in fighting against the ever present and evolving DDoS threat. In conclusion, through constant innovation in ML methodologies and by mitigating many unresolved issues, robust and intelligent systems can be designed to protect digital infrastructures against intelligent cyber-attacks and guarantee the resilience of critical network services.

References

1. Saghezchi, F. B., Mantas, G., Violas, M. A., de Oliveira Duarte, A. M., & Rodriguez, J. (2022). Machine learning for DDoS attack detection in industry 4.0 CPPSs. *Electronics*, *11*(4), 602.
2. Corrêa, J. H., Ciarelli, P. M., Ribeiro, M. R., & Villaça, R. S. (2021). Ml-based ddos detection and identification using native cloud telemetry macroscopic monitoring. *Journal of Network and Systems Management*, *29*, 1-28.
3. Sudusinghe, C., Charles, S., & Mishra, P. (2021, October). Denial-of-service attack detection using machine learning in network-on-chip architectures. In *Proceedings of the 15th IEEE/ACM International Symposium on Networks-on-Chip* (pp. 35-40).
4. Syed, N. F., Baig, Z., Ibrahim, A., & Valli, C. (2020). Denial of service attack detection through machine learning for the IoT. Journal of Information and Telecommunication, 4(4), 482-503.
5. Islam, U., Muhammad, A., Mansoor, R., Hossain, M. S., Ahmad, I., Eldin, E. T., ... & Shafiq, M. (2022). Detection of distributed denial of service (DDoS) attacks in IOT based monitoring system of banking sector using machine learning models. *Sustainability*, *14*(14), 8374.
6. Doshi, R., Apthorpe, N., & Feamster, N. (2018, May). Machine learning ddos detection for consumer internet of things devices. In *2018 IEEE Security and Privacy Workshops (SPW)* (pp. 29-35). IEEE.
7. Gadze, J. D., Bamfo-Asante, A. A., Agyemang, J. O., Nunoo-Mensah, H., & Opare, K. A. B. (2021). An investigation into the application of deep learning in the detection and mitigation of DDOS attack on SDN controllers. *Technologies*, *9*(1), 14.

8. Bindra, N., & Sood, M. (2019). Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset. *Automatic Control and Computer Sciences*, *53*(5), 419-428.
9. Lima Filho, F. S. D., Silveira, F. A., de Medeiros Brito Junior, A., Vargas-Solar, G., & Silveira, L. F. (2019). Smart detection: an online approach for DoS/DDoS attack detection using machine learning. *Security and Communication Networks*, *2019*(1), 1574749.
10. Sangodoyin, A. O., Akinsolu, M. O., Pillai, P., & Grout, V. (2021). Detection and classification of DDoS flooding attacks on software-defined networks: A case study for the application of machine learning. *IEEE Access*, *9*, 122495-122508.
11. Bawany, N. Z., Shamsi, J. A., & Salah, K. (2017). DDoS attack detection and mitigation using SDN: Methods, practices, and solutions. *Arabian Journal for Science and Engineering*, *42*, 425–441. https://doi.org/10.1007/s13369-016-2182-6
12. Despaux, F. (2015). *Modelling and evaluation of the end to end delay in WSN* (Doctoral dissertation, Université de Lorraine).
13. Dong, S., & Sarem, M. (2019). DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks. *IEEE Access*, *8*, 5039–5048. https://doi.org/10.1109/ACCESS.2019.2895658
14. Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019, October). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In *2019 International Carnahan Conference on Security Technology (ICCST)* (pp. 1–8). IEEE.
15. Yuan, J., & Mills, K. (2005). Monitoring the macroscopic effect of DDoS flooding attacks. *IEEE Transactions on Dependable and Secure Computing*, *2*(4), 324–335.
16. Zhu, J., Dai, Q., Deng, Y., Zhang, A., Zhang, Y., & Zhang, S. (2018). Indirect damage of urban flooding: Investigation of flood-induced traffic congestion using dynamic modeling. *Water*, *10*(5), 622. https://doi.org/10.3390/w10050622
17. Analysis of features dataset for DDoS detection by using ASVM method on software defined networking [Figure]. (n.d.). *ResearchGate*. Retrieved January 21, 2025, from https://www.researchgate.net/figure/Architecture-of-Distributed-Denial-of-Service-DDoS-attack_fig1_340530507
18. Ussatova, O., Zhumabekova, A., Begimbayeva, Y., Matson, E. T., & Ussatov, N. (2022). Comprehensive DDoS Attack Classification Using Machine Learning Algorithms. *Computers, Materials & Continua*, *73*(1).
19. Miglani, A., & Kumar, N. (2019). Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges. *Vehicular Communications*, *20*, 100184.