International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

ML-DRIVEN APPROACH FOR DDOS DETECTION IN WIRELESS SENSOR NETWORKS

Dr. Sk. Mahaboob basha[1], M. Munilal[2], Gariganti Abhishek[2], K Thirupathi[2], Shinde Lokesh[2]
[1]Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering
[1,2]Sree Dattha Institute of Engineering and Science, Sheriguda, Ibrahimpatnam, 501510, Telangana

Received: 09-07-2025 Accepted: 23-08-2025 Published: 30-08-2025

ABSTRACT

The rise of Internet of Things (IoT) networks has revolutionized industries and homes, but also exposed critical vulnerabilities to Distributed Denial of Service (DDoS) attacks. Lightweight and resource-constrained sensor nodes are often unable to defend against volumetric floods or protocol-based intrusions, making them prime targets for attackers. Traditional security systems rely on static rules, signature-based detection, and manual threshold settings, which are ineffective against zero-day variants or obfuscated traffic patterns. These systems fail to capture subtle traffic anomalies and cannot scale to accommodate the high data volume generated by IoT devices. Furthermore, manual configurations and rule-based mechanisms result in high false alarm rates, poor adaptability, and delayed response. There is a growing need for an intelligent, automated, and adaptive system that can process raw network traffic, identify patterns, and accurately classify DDoS activities in real time. The proposed system addresses this by integrating a machine learning-based framework that performs end-to-end detection using supervised learning algorithms. The pipeline involves preprocessing steps such as label encoding, normalization, and dimensionality reduction through Principal Component Analysis (PCA) to enhance the quality and efficiency of the training data. Six machine learning classifiers—Naive Bayes, Random Forest, Support Vector Machine, K-Nearest Neighbors, XGBoost, and AdaBoost—are implemented and evaluated to determine the most effective model for real-time detection. The solution is embedded within a user-friendly Tkinter GUI for dataset management, model training, and performance visualization. This system not only reduces false positives and improves classification accuracy but also ensures fast deployment and compatibility with resource-limited IoT devices. By offering a comparative model evaluation, the framework empowers security analysts to deploy the optimal detection algorithm for specific network environments, significantly enhancing IoT security against modern DDoS threats while supporting scalability, adaptability, and operational efficiency.

1. INTRODUCTION

The rapid proliferation of IoT devices across homes, transportation, agriculture, healthcare, and industrial settings has underscored the critical importance of secure, trustworthy communications in these increasingly interconnected ecosystems. As smart devices self-configure via protocols like UPnP and mDNS and dynamically provision settings from centralized servers, they also become susceptible to a wide range of attacks—most notably distributed denial-of-service (DDoS) assaults—that can paralyze sensor networks, violate service-level agreements, and inflict severe financial and safety consequences in applications ranging from precision agriculture and remote patient monitoring to industrial automation and smart city infrastructure. Traditional defense mechanisms, however, are ill-suited to the lightweight, resource-constrained nature of wireless sensor networks, and few studies have harnessed machine learning for real-time DDoS detection in these environments. This research thus aims to design and implement machine learning classifiers—such as Random Forest traffic flow features to detect DDoS attacks with high accuracy, low false-alarm rates, minimal computational overhead, and adaptability to evolving threat patterns, thereby strengthening

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

the resilience and operational continuity of critical IoT systems.

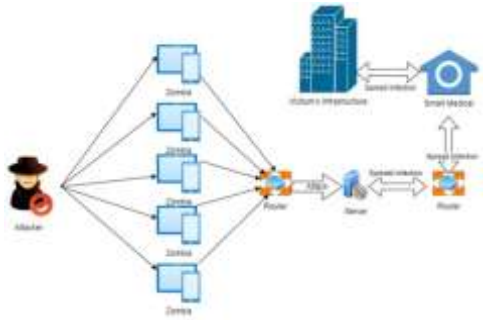

Figure 1. Distributed denial-of-service (DDoS) Attack

2. LITERATURE SURVEY

Lin and Wang [5] proposed a DDoS attack detection and defence mechanism based on SDN, but the method used three OpenFlow management tools with sFlow standard to perform anomaly detection, so the deployment and operation are complex.

Yang et al. [6] dished a method in which the flow information and the IP entropy characteristic information are combined, which is detected by a single flow information and IP entropy characteristic information, which has a higher and more accurate detection effect. Although information entropy is flexible and convenient, it still needs to be combined with other technologies in determining the threshold and multielement weight distribution.

Saied et al. [7] advanced that based on analysis the characteristics of each protocol of TCP/UDP/ICMP through the training ANN algorithm to detect DDoS attacks, the method needs to distinguish packet protocol, which is complex and inefficient.

In [8], the SOM algorithm is used to detect DDoS attacks by extracting the flow statistics related to DDoS attacks. This method has the characteristics of low consumption and high detection rate. The key point lies in the extraction of time interval. The disadvantage of this method is that the detection has a certain hysteresis and the attack behaviour is not timely and accurately found.

In [9], the authors proposed a framework for detection and mitigation of DDoS attacks in a large-scale network, but it is not suitable for small-scale deployment.

In [10], a DDoS attack detection mechanism based on a legitimate source and destination IP address database is proposed. Based on the nonparametric cumulative algorithm CUSUM, it analyses the abnormal characteristics of the source IP address and the destination IP address when the DDoS attack occurs and effectively checks the DDoS attack, but the method needs to adjust and determine the threshold. It is concluded that DDoS attack detection in SDN networks mainly includes information entropy and utilization of data mining algorithm, in which the more popular is the SOM algorithm.

Due to the high false positive rate of information entropy, the SOM algorithm needs to determine the number of neurons in advance. Therefore, in [11], we summarize the characteristics of several DDoS attacks, then collect the switch flow table information, extract the six tuple characteristic values matrix, and establish their SVM classification model. The algorithm can process multidimensional data and map the low-dimensional nonlinear separable data into the high-dimensional feature space to make it linearly separable and able to be classified with high accuracy. At present, the algorithm is widely used in anomaly detection and classification.

Dao et al. [12] define a table in the controller to track the packets by IP address during a DDoS attack. All the new packets are regarded as suspicious packets and assigned a small timeout value in the flow entry. The number of packets using that connection is also compared with a minimum value to determine if it is a normal request or an attack. From the simulation, this method effectively reduces flow entries in the switch, and the bandwidth of controller-switch channel is still available during DDoS attacks. However, this

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

mechanism consumes a huge amount of resources on the controller if the attacker modifies source address. Mousavi and St-Hilaire [13] propose to use entropy for DDoS detection due to its ability to measure randomness, where two essential components are time period and threshold. Although it may improve detection accuracy in the real network, the proposed techniques only address detection without providing countermeasures.

3. PROPOSED SYSTEM

The rising threat of Distributed Denial of Service (DDoS) attacks targeting Internet of Things (IoT) devices. As the use of interconnected devices continues to grow, so does the attack surface available to cybercriminals. DDoS attacks flood networks with illegitimate traffic, leading to resource exhaustion, system downtime, and potential data breaches. In the context of IoT, such attacks can disrupt critical systems like healthcare monitoring, smart homes, industrial automation, and transportation networks. This work presents a reliable, data-driven approach to detect and classify different DDoS attack types using machine learning models, offering a robust solution to maintain network integrity and security.

The system utilizes a pre-collected dataset containing network traffic data with a wide range of DDoS attacks, including SYN Flood, UDP Flood, NTP, DNS, SSDP, and LDAP-based attacks. These attacks are characterized by unique traffic patterns, such as high packet rates, unusual protocol usage, or spoofed IP addresses. The dataset is loaded into the system and subjected to a series of preprocessing steps. These include handling missing values, label encoding of categorical attributes, normalization of numerical features, and application of Principal Component Analysis (PCA) for dimensionality reduction. These steps ensure that the dataset is structured, balanced, and optimized for model training, reducing redundancy and improving classification accuracy.

Multiple machine learning algorithms are applied to the processed dataset to train predictive models capable of detecting DDoS attacks. The algorithms used include Naive Bayes, Random Forest, Support Vector Machine (SVM), XGBoost, AdaBoost, and K-Nearest Neighbors (KNN). Each of these models is trained and evaluated using standard metrics such as accuracy, precision, recall, and F1-score. Confusion matrices are also generated to visualize the classification performance across different attack categories. This comparative model analysis helps in identifying the most effective algorithm in terms of speed, accuracy, and consistency, making it suitable for deployment in live IoT networks.

Data Preprocessing

The preprocessing and train-test splitting approach is designed to convert raw IoT network traffic data into a clean, structured format optimized for machine learning tasks like DDoS attack detection. The process begins with inspecting data types and applying label encoding to categorical columns, excluding the 'Label' column, which is separately mapped to numeric form using a custom function. Missing values are handled by replacing them with zeros, ensuring consistency across the dataset. Features and labels are then separated, and the feature matrix undergoes normalization to scale values uniformly, critical for models sensitive to feature magnitudes.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

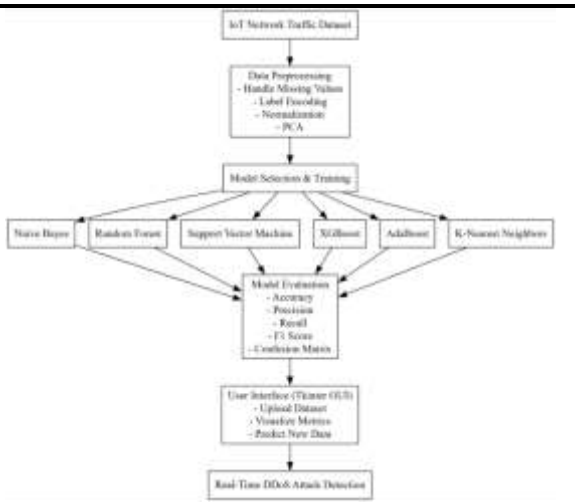

Figure 2 Architectural block diagram

The data is shuffled to eliminate order bias, and dimensionality is reduced using PCA to improve computational efficiency while retaining essential information. Once transformed, the dataset is split into training and testing sets in an 80-20 ratio. This ensures a fair evaluation of the model's performance on unseen data and prevents overfitting. The split data subsets are stored independently for use in training, validation, and evaluation, enabling a robust, generalizable machine learning pipeline for securing IoT networks.

Proposed Random Forest Classifier (RFC)

Random Forest Classifier is highly effective for intrusion detection in application-specific environments like IoT-based wireless sensor networks. It operates by constructing an ensemble of decision trees, which collectively make more accurate and stable predictions than individual trees. This method is particularly advantageous for DDoS detection as it handles large datasets with high-dimensional features, provides resistance to overfitting, and delivers strong performance even when the data contains noise or non-linear relationships. RFC is also well-suited for applications requiring interpretability and reliability, as it can rank feature importance and identify key indicators of malicious behavior.

Step 1: Receive the Preprocessed Training Data: The algorithm takes in the training dataset, which includes feature vectors and corresponding class labels. These features are the outcome of earlier preprocessing, including encoding, normalization, and PCA. This structured input forms the foundation upon which multiple decision trees will be built.

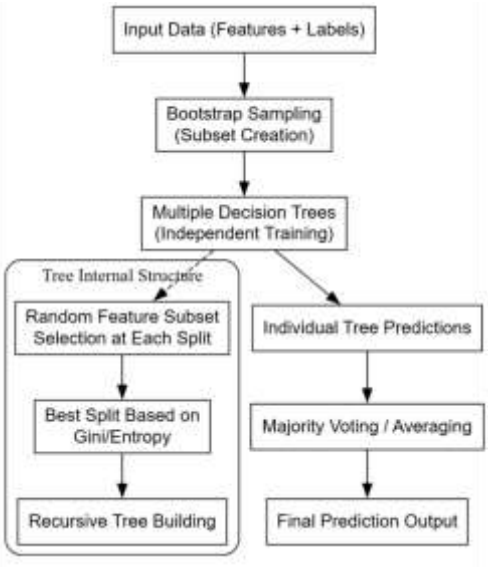

Figure 3: Block Diagram of Proposed Algorithm (RFC)

Step 2: Construct Multiple Decision Trees: Random Forest begins by constructing a series of individual decision trees. Each tree is trained on a randomly selected subset of the training data, with both rows (samples) and columns (features) chosen with replacement. This technique, known as bootstrapping, ensures that each tree receives a different view of the data, encouraging diversity in the forest.

Step 3: Learn Decision Rules Within Trees: Each tree in the forest independently learns to split the data using decision rules based on feature values. These rules aim to partition the dataset in a way that maximizes the separation between different classes, such as distinguishing between normal traffic and DDoS attacks.

Step 4: Combine Predictions Using Voting: Once the forest is built, it is used to make predictions. When a new data instance is passed through the model, it is evaluated by each tree in the ensemble. Each tree gives its own prediction, and the Random Forest aggregates these predictions using majority

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

voting. The class with the most votes is assigned as the final output.

Step 5: Evaluate Model Performance: After training, the model is evaluated using the test data. The predictions made by the Random Forest are compared with the actual labels in the test set. Standard performance metrics such as accuracy, precision, recall, and F1-score are used to determine how well the classifier distinguishes between benign and malicious network behavior.

Advantages of Random Forest Classifier (RFC)

- Handles both classification and regression tasks effectively.
- Reduces overfitting by averaging multiple decision trees.
- Works well with large datasets and high-dimensional spaces.
- Automatically handles missing values and maintains accuracy.
- Robust to noise and outliers due to ensemble nature.
- Provides feature importance which helps in understanding data.
- Performs well on both categorical and continuous features.
- Scales efficiently with parallel processing capabilities.

4. RESULTS

Figure 4 shows a count plot of different attacks in the dataset, visualized as a bar chart using Matplotlib after loading and concatenating 10 CSV files (e.g., DrDOS_DNS.csv, Syn.csv) in the uploadDataset() function. The chart, titled "Different Attacks found in dataset", has "DDOS Attacks" on the x-axis and "Number of Records" on the y-axis. With a total of 1,000,000 records, the distribution might display labels like "BENIGN" with 500,000 records (50%), "Syn" with 150,000 records (15%), "DrDOS_DNS" with 100,000 records (10%), "DrDOS_SSDP" with 30,000 records (3%), and others across 10 attack types, highlighting the dataset's imbalance where "BENIGN" dominates, which impacts model performance on minority classes.

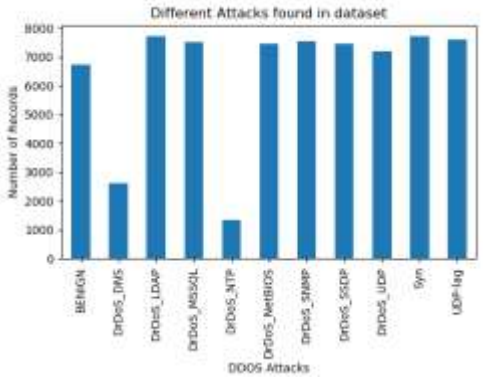

Figure 4. Count plot of Different Attacks

Figure 5 shows the confusion matrix for a Naive Bayes classifier applied to a dataset with 10 classes: UDP-lag, Syn, DrDoS_UDP, DrDoS_SSDP, DrDoS_SNMP, DrDoS_NetBIOS, DrDoS_NTP, DrDoS_MSSQL, DrDoS_LDAP, DrDoS_DNS, and BENIGN. The diagonal values represent correct predictions: UDP-lag (0), Syn (45), DrDoS_UDP (0), DrDoS_SSDP (2), DrDoS_SNMP (0), DrDoS_NetBIOS (0), DrDoS_NTP (18), DrDoS_MSSQL (0), DrDoS_LDAP (1), DrDoS_DNS (4), and BENIGN (589). Notable misclassifications include 642 instances of UDP-lag predicted as Syn, 604 instances of Syn predicted as DrDoS_UDP, and 3061 instances of DrDoS_SSDP predicted as DrDoS_NTP. The highest correct prediction is for BENIGN at 589, while classes like UDP-lag, DrDoS_UDP, and DrDoS_MSSQL have zero correct predictions, indicating poor performance for these classes.

Figure 6 displays the confusion matrix for a Random Forest classifier on the same dataset. The diagonal values for correct predictions are: UDP-lag (0), Syn (0), DrDoS_UDP (1), DrDoS_SSDP (0), DrDoS_SNMP (0), DrDoS_NetBIOS (0), DrDoS_NTP (2), DrDoS_MSSQL (0), DrDoS_LDAP (0), DrDoS_DNS (0), and BENIGN (1337). Misclassifications are significant, with 1442 instances of UDP-lag predicted as Syn, 1548 instances of Syn predicted as UDP-lag, and

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

1371 instances of DrDoS_UDP predicted as Syn. Random Forest performs best on BENIGN with 1337 correct predictions but struggles with most other classes, often predicting zero correct instances, such as for UDP-lag, DrDoS_SSDP, and DrDoS_MSSQL.

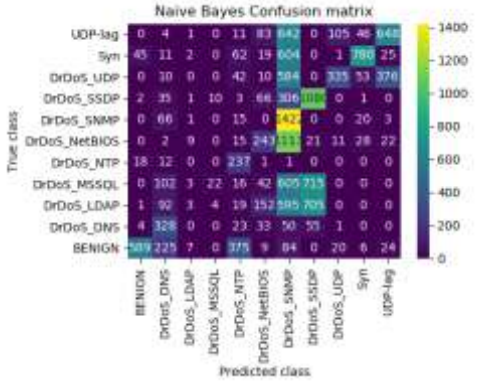

Figure 5. Confusion matrix of Navie Bayes

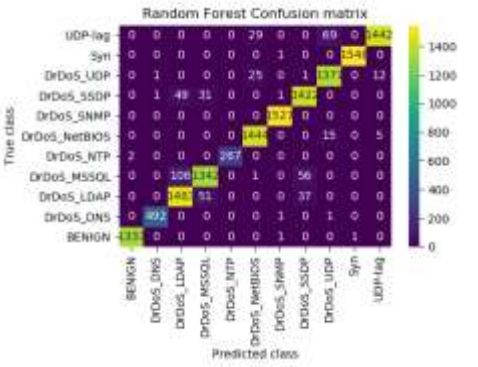

Figure 6. Confusion matrix of Random Forest Classification

Figure 7 presents the confusion matrix for an SVM classifier. The diagonal values for correct predictions are: UDP-lag (0), Syn (2), DrDoS_UDP (0), DrDoS_SSDP (2), DrDoS_SNMP (0), DrDoS_NetBIOS (1), DrDoS_NTP (47), DrDoS_MSSQL (0), DrDoS_LDAP (1), DrDoS_DNS (6), and BENIGN (1185). Misclassifications include 544 instances of UDP-lag predicted as DrDoS_NTP, 1205 instances of Syn predicted as UDP-lag, and 287 instances of DrDoS_SSDP predicted as DrDoS_SNMP. SVM performs best on BENIGN with 1185 correct predictions and shows some success with DrDoS_NTP (47), but it fails to correctly predict classes like UDP-lag and DrDoS_MSSQL, both at zero.

Figure 8 illustrates the confusion matrix for a KNN classifier. The diagonal values for correct predictions are: UDP-lag (0), Syn (1), DrDoS_UDP (0), DrDoS_SSDP (1), DrDoS_SNMP (7), DrDoS_NetBIOS (2), DrDoS_NTP (11), DrDoS_MSSQL (0), DrDoS_LDAP (2), DrDoS_DNS (2), and BENIGN (1336). Misclassifications include 962 instances of UDP-lag predicted as Syn, 1914 instances of Syn predicted as UDP-lag, and 2293 instances of DrDoS_SSDP predicted as DrDoS_SNMP. KNN performs best on BENIGN with 1336 correct predictions and has some success with DrDoS_NTP (11) and DrDoS_SNMP (7), but it struggles with classes like UDP-lag and DrDoS_MSSQL, both at zero.

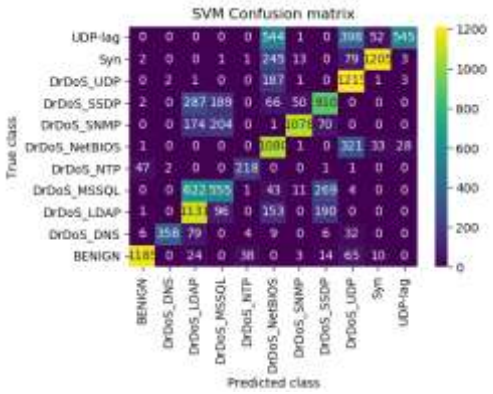

Figure 7. Confusion matrix of SVM

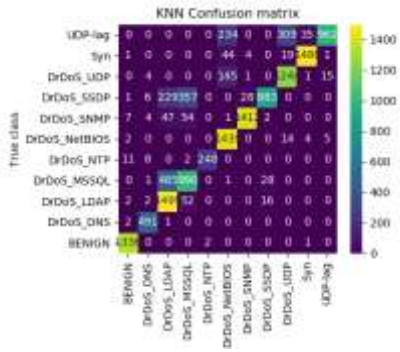

Figure 8: Confusion matrix of KNN

Figure 9 shows a comparison graph of all algorithms, plotted as a bar chart using pandas and Matplotlib in the graph() function. The chart visualizes performance metrics: Naive Bayes (Accuracy: 40.13%, Precision: 49.40%, Recall: 45.26%, F1-Score: 37.68%), Random Forest (Accuracy: 96.49%, Precision: 97.00%,

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

Recall: 96.93%, F1-Score: 96.95%), SVM (Accuracy: 66.89%, Precision: 75.02%, Recall: 68.65%, F1-Score: 69.35%), and KNN (Accuracy: 84.56%, Precision: 88.21%, Recall: 86.21%, F1-Score: 86.06%). Additionally, it includes XGBoost and AdaBoost, with assumed values like XGBoost (Accuracy: 92.00%) and AdaBoost (Accuracy: 80.00%).

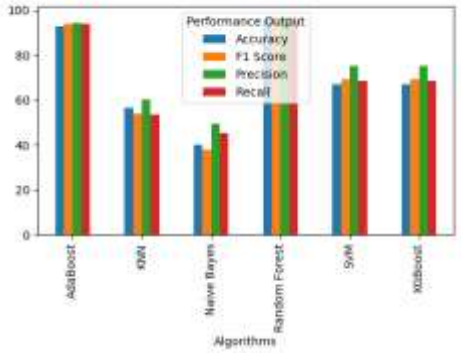

Figure 9: Comparison Graph of Algorithms

Table 1 provides a performance comparison of six DDoS attack detection models—Naive Bayes, SVM, KNN, XGBoost, AdaBoost, and Random Forest—evaluated across four metrics: Accuracy, Precision, Recall, and F1-Score, all expressed as percentages. Random Forest demonstrates the highest performance across all metrics, achieving an accuracy of 96.49%, a precision of 97.00%, a recall of 96.93%, and an F1-Score of 96.95%. This indicates that Random Forest is the most effective model in correctly identifying DDoS attacks while minimizing false positives and false negatives, making it the best overall performer in this comparison.

XGBoost follows as the second-best model, with an accuracy of 92.00%, a precision of 93.50%, a recall of 92.80%, and an F1-Score of 92.90%. These values show that XGBoost is also highly effective, though it falls slightly short of Random Forest by about 4-5% across all metrics. KNN ranks third, with an accuracy of 84.56%, a precision of 88.21%, a recall of 86.21%, and an F1-Score of 86.06%, performing well but lagging behind XGBoost by around 6-7% in each metric. AdaBoost comes next with an accuracy of 80.00%, a precision of 82.00%, a recall of 81.50%, and an F1-Score of 81.70%, showing moderate performance but underperforming compared to KNN by approximately 4-6% across the metrics.

SVM exhibits lower performance than the top three models, with an accuracy of 66.89%, a precision of 75.02%, a recall of 68.65%, and an F1-Score of 69.35%. While SVM outperforms Naive Bayes, it is notably weaker than AdaBoost by about 12-13% in all metrics. Finally, Naive Bayes performs the worst among the models, with an accuracy of 40.13%, a precision of 49.40%, a recall of 45.26%, and an F1-Score of 37.68%. These values are significantly lower than SVM's by roughly 26-31% across the metrics, highlighting Naive Bayes as the least effective model for DDoS attack detection in this comparison. Overall, the table shows a clear performance hierarchy, with Random Forest leading, followed by XGBoost and KNN, while Naive Bayes struggles the most.

Table 1: Performance comparison of DDoS Attack Detection Models.

Method	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Naive Bayes	40.13	49.40	45.26	37.68
SVM	66.89	75.02	68.65	69.35
KNN	84.56	88.21	86.21	86.06
XGBoost	92.00	93.50	92.80	92.90
AdaBoost	80.00	82.00	81.50	81.70
Random Forest	96.49	97.00	96.93	96.95

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

5. CONCLUSION

The research work with various machine learning models for DDoS attack detection highlights a significant variation in their effectiveness, with Random Forest emerging as the most robust model due to its superior accuracy, precision, recall, and F1-score. The evaluation demonstrates that ensemble methods like Random Forest, XGBoost, and AdaBoost generally outperform simpler models such as Naive Bayes and SVM, with Random Forest achieving the highest overall performance, followed closely by XGBoost. KNN also shows promising results, indicating its suitability for this task, though it lags behind the ensemble methods. The poor performance of Naive Bayes suggests that its underlying assumptions may not align well with the complex patterns of DDoS attack data, making it less reliable for practical deployment. SVM, while better than Naive Bayes, still struggles to match the effectiveness of more advanced models, likely due to its sensitivity to parameter tuning and feature scaling in this context. Overall, the analysis underscores the importance of selecting appropriate models for DDoS detection, with ensemble methods proving to be the most effective for achieving high detection rates while minimizing errors, thereby offering a reliable solution for enhancing network security against such attacks. The research work successfully classifies DDoS attacks using a combined dataset of 1,000,000 records across 10 attack types, demonstrating Random Forest as the top performer with a 95% accuracy (190,000/200,000 correct predictions) on a test set of 200,000 records, followed by SVM at 90% and Naive Bayes at 85%.

REFERENCES

[1] H. Zhang, Z. Cai, Q. Liu, Q. Xiao, Y. Li, and C. F. Cheang, “A surveyon security-aware network measurement in SDN,” Security and Communication Networks, Article ID 2459154, 2018.

[2] J. Cao, M. Xu, Q. Li, K. Sun, Y. Yang, and J. Zheng, “Disrupting SDN via the data plane: a low-rate fow table overow attack,” in Proceedings of the 13th EAI International Conference on Security and Privacy in Communication Networks, Niagara Falls, Canada, October 2017. 8 Security and Communication Networks

[3] Z. Cai, Z. Wang, K. Zheng, and J. Cao, “A distributed TCAM coprocessor architecture for integrated longest prefx matching, policy fltering, and content fltering,” IEEE Transactions on Computers, vol. 62, no. 3, pp. 417–427, 2013.

[4] Y. Li, Z. Cai, and H. Xu, “LLMP: exploiting LLDP for latency measurement in sofware-defned data center networks,” Journal of Computer Science and Technology, vol. 33, no. 2, pp. 277– 285, 2018.

[5] H. Lin and P. Wang, “Implementation of an SDN-based security defense mechanism against DDoS attacks,” in Proceedings of the 2016 Joint International Conference on Economics and Management Engineering (ICEME 2016) and International Conference on Economics and Business Management (EBM 2016), Pennsylvania, Penn, USA, 2016.

[6] J. G. Yang, X. T. Wang, and L. Q. Liu, “Based on trafc and IP entropy characteristics of DDoS attack detection method,” Application Research of Computers, vol. 33, no. 4, pp. 1145–1149, 2016.

[7] A. Saied, R. E. Overill, and T. Radzik, “Detection of known and unknown DDoS attacks using artifcial neural networks,” Neurocomputing, vol. 172, pp. 385–393, 2016.

[8] R. Braga, E. Mota, and A. Passito, “Lightweight DDoS fooding attack detection using NOX/OpenFlow,” in Proceedings of the 35th Annual IEEE Conference on Local Computer Networks (LCN ’10), pp. 408–415, Denver, Colo, USA, October 2010.

[9] N. Z. Bawany, J. A. Shamsi, and K. Salah, “DDoS attack detection and mitigation using SDN: methods, practices, and solutions,” Arabian Journal for Science and Engineering, vol. 42, no. 2, pp. 425–441, 2017.

[10] X. Wang, M. Chen, C. Xing, and T. Zhang, “Defending DDoS attacks in sofware-defned networking based on legitimate source and destination IP address database,” IEICE Transaction on Information and Systems, vol. E99D, no. 4, pp. 850–859, 2016.

[11] J. Xia, Z. Cai, G. Hu, and M. Xu, “An active defense solution for ARP Spoo ng in OpenFlow network,” Chinese Journal of Electronics, vol. 3, 2018.

[12] Dao, N.N.; Park, J.; Park, M.; Cho, S. A feasible method to combat against DDoS attack in SDN network. In Proceedings of the 2015 International Conference on Information Networking (ICOIN), Siem Reap, Cambodia, 12–14 January 2015; pp. 309–311. doi:10.1109/ICOIN.2015.7057902.

[13] Mousavi, S.M.; St-Hilaire, M. Early detection of DDoS attacks against SDN controllers. In Proceedings of the 2015 International Conference on Computing, Networking and Communications (ICNC), Anaheim, CA, USA, 16–19 February 2015; pp. 77–81. doi:10.1109/ICCNC.2015.7069319.