Bima Journal of Science and Technology, Vol. 8(4A) Dec, 2024 ISSN: 2536-6041

DOI: 10.56892/bima.v8i4.1156

Intrusion Detection of Distributed Denial of Service Attack in Internet of Things Network Using Adaboost and Support Vector Machine

Abdulkadir Abdulkadir Baba, Saka Kayode Kamil, Gbolagade Morufat Damola, Abdulkadir Khalimat and Bello Kamoru Atanda

Depaertment of Computer Science, Al-Hikmah University Ilorin, Kawara State, Nigeria

Corresponding Author: kamilsaka67@gmail.com

ABSTRACT

Distributed denial-of-service (DDoS) attacks pose a significant threat to computer networks and systems by disrupting services through the saturation of targeted systems with traffic from multiple sources. Real-time detection of these attacks has become a critical cyber security task. However, current DDoS attack detection methods suffer from high false positive rates and limited ability to capture the complex patterns of attack traffic. This research developed an effective non-real-time DDoS attack detection system for IoT networks. The methods chosen for this research, including Support Vector Machines (SVM), Adaptive Boosting (AdaBoost), and feature selection techniques i.e Mutual Information (MI) and Recursive Feature Elimination with Cross-Validation (RFECV) on CICIDS2019 dataset. The SVM-AdaBoost classifier achieved an accuracy of approximately 99.9% the precision is 99.5% Recall 99.1%, F1score 99.3% and AUC-ROC 100%. While the training time display an average time of 1.2031sec. The results of this study suggest that security professionals and researchers should consider adopting ensemble methods like AdaBoost, especially when combined with robust base learners such as SVM, in the development of intrusion detection systems for IoT networks

Keywords: Distributed denial-of-service (DDoS), IoT networks, Support Vector Machines (SVM), Adaptive Boosting (AdaBoost), CICIDS2019 dataset.

INTRODUCTION

The swift growth of Internet of Things (IoT) networks, which are comprised of billions of linked devices, has revolutionized several sectors by permitting instantaneous data gathering and automation (Sambangi & Gondi, 2024). But with greater connection comes new security risks as well, such as Distributed Denial of Service (DDoS) attacks, which try to overload network capacity and interfere with regular business as usual (Elliot, 2021; Dasari, & Devarakonda, 2021). IoT devices are particularly susceptible to these kinds of assaults because of their constrained memory, processing capacity, and security mechanisms (Ashton, 2023). Traditional Intrusion Detection Systems (IDS), such as signature-based and anomaly-based techniques, have failed to keep up with the increasing sophistication and size of DDoS assaults (Zhang, Hu & Ji, 2023). This has led to low detection accuracy and higher false-positive rates.

Cyber-attacks or intrusion may be malicious or non-malicious. A Distributed Denial of Service (DDoS) attack is an example of a malicious intrusion. A Distributed Denial of Service attack is a coordinated cyber attack where multiple compromised systems flood a target system with an enormous volume of traffic, aiming to overwhelm its resources and incapacitate its ability to serve legitimate users (Sambangi & Gondi, 2020; Arora, Yadav, & Sharma, 2018). In a DDoS attack, an attacker commands a botnet, a

DOI: 10.56892/bima.v8i4.1156

collection of infected devices, to produce a massive influx of traffic, and this decentralized nature makes it difficult to control (Sambangi & Gondi, 2020).

Several machine learning (ML) methods have been put out to improve IoT intrusion detection. Settings, several methods encounter difficulties because of the IoT traffic data's high dimensionality and complexity (Mendy and Elliot, 2023). It is now critical to have detection systems that are scalable, accurate, and efficient. Adaptive Boosting and Support Vector Machines (SVM) are two of the most robustly performing machine learning classification algorithms. Nevertheless, there is still much to learn about applying these algorithms to IoT-based intrusion detection, particularly for DDoS attacks (Awad & Fraihat, 2023). Moreover, little study has been done on the application of feature selection techniques to enhance the speed and precision of these models (Richard and Micheal, 2023). This research attempts to close the gap by maximizing the application of machine learning to enhance accuracy in non-real-time analytical contexts, current intrusion detection techniques can be improved

REVIEW OF RELATED WORKS

Abolarinwa et al., (2024) investigated the use of ensemble machine learning techniques in the development of a distributed denial of service detection model. This work uses ensemble machine learning (ML) models that combine Bagging, Boosting, and Stacking techniques. The agile software development methodology was used for the implementation in order to facilitate changes at each stage. The HTML, CSS, and JavaScript frameworks were used in the development of the user interface. Several assessment metrics were used to evaluate the ensemble models. The Bagging Ensemble method fared better than the other models, with an approximate F1-score of 95.61%, 97% precision, 94.88% recall, and 99.5% accuracy. The experimental results showed that the bagging ensemble approach is recommended for constructing a DDoS attack detection model. Future research should concentrate on lowering the feature by using machine learning algorithms for feature selection, as this work has the issue of having large dimensionality features.

Alamgir & Saiful (2024). This study suggests a hybrid feature selection method in conjunction with ensemble-based classifiers as an improved method for identifying DDoS attacks. Several decision trees are combined in the ensemble-based method to improve classification accuracy, decrease overfitting, and strengthen the model. Principal component analysis, mutual information, and correlation analysis are used in the feature selection process to determine which attributes are most helpful for attack detection. The suggested model is tested on a variety of DDoS attack detection datasets, and experimental results show that it outperforms current methods in terms of accuracy, recall, precision, f1-score, and false positive rate, among other assessment metrics. The suggested method is a viable option for DDoS attack detection because it achieves about 96% accuracy, 96% true positive rate, and 0% error rate. Future research should apply another approach to increase the model's accuracy.

Yakub et al., (2022) suggests an intrusion detection system (ML-IDS) that uses machine learning to identify assaults on Internet of Things networks. The UNSW-NB15 dataset was subjected to feature scaling in the first stage of this study using the Minimum-maximum (min–max) idea of normalization in order to minimize

DOI: 10.56892/bima.v8i4.1156

information leakage on the test data. The next step involved using Principal Component Analysis (PCA) to reduce dimensionality. The experimental results of our findings were evaluated in terms of validation data- set, accuracy, the area under the curve, recall, F1, precision, kappa, and Mathew correlation coefficient (MCC). The findings were also benchmarked with the existing works, and the results were competitive with an accuracy of 99.9% and MCC of 99.97%. Future work should increase the number of features in the dataset to make the system more robust.

Mahrukh et al., (2023) investigated the use of deep learning algorithms for network traffic detection of distributed denial of service attacks. This research uses deep learning models, such as gradient recurrent units (GRU), long short-term memory (LSTM), and recurrent neural networks (RNN), to identify DDoS attacks on the CICIDS2019 dataset. The experimental findings show that models function similarly on the CICIDS2019 dataset, with an accuracy score of 0.99, but there is a difference in execution time, with GRU showing less execution time than those of RNN and LSTM.

Oyong, Ekong & Obot (2023) examined the use of KNN, SVM base classifiers, and the Adaboost algorithm for dynamic analysis of malware intrusion in mobile devices. This research work is keying into the fight against malware intrusion by designing and developing an intrusion detection system (IDS) using ensemble learning, boosting. Adaboost algorithm trains base classifiers (KNN and SVM) using network security laboratory-knowledge discovery in databases (NSL-KDD) dataset to build a more formidable classifier that will detect malware intrusion in mobile devices using cloud technology. The result obtained in this combination technique is 91.4% accurate with a bias (standard deviation) as low as 2.7%.

MATERIALS AND METHODS

This chapter explains in particular all the research processes required to achieve all objectives listed in the introduction section. The methods and models were chosen for this study, including Support Vector Machines (SVM), Adaptive Boosting (AdaBoost), and feature selection techniques i.e Mutual Information (MI) and Recursive Feature Elimination with Cross-Validation (RFECV), were selected based on their ability to handle the complexity and high dimensionality of IoT network data while optimizing detection accuracy and computational efficiency. To accomplish the objectives, the general research methods are outlined in Figure 1.

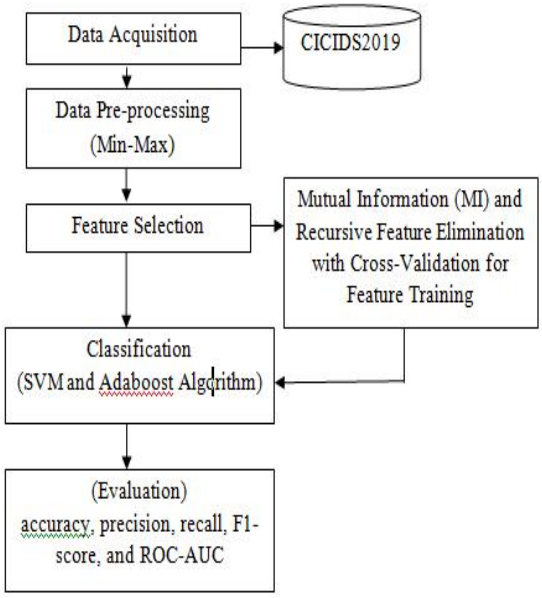

Figure 1: Research Framework.

Data Collection

Data collection is a crucial step in this study as it provides the necessary network traffic information required to develop and train

DOI: 10.56892/bima.v8i4.1156

machine learning models capable of detecting Distributed Denial of Service (DDoS) attacks (Wu et al., 2023). The CICIDS2019 dataset is utilized in this study, offering a wide array of network traffic attributes collected during both attack and normal network behavior.

Data Preprocessing

Given the range of feature values, including byte rates, packet sizes, and flow lengths, normalizing the dataset is essential. Min-Max Scaling, which scales all features to a range between 0 and 1, is used to remedy this. This method is especially helpful for distance-based algorithms that depend on the size of the input data. Through the use of Min-Max normalization, this study was able to maintain the associations between data points while guaranteeing that every feature contributes equally to the model and that no single feature dominates the learning process. All features are scaled to a uniform range, which can enhance the performance of feature selection algorithms.

Mathematical Equation for Min-Max Normalization

The Min-Max Scaling formula is:

$$X' = \frac{X - \min(x)}{\max(x) - \min(x)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Where:

- x is the original value of the feature,
- min(x) is the minimum value of the feature in the dataset,
- max(x) is the maximum value of the feature in the dataset,
- x' is the scaled value of the feature.

Steps in Applying Min-Max Scaling:

Identify Features for Scaling: Determine which features require scaling. In most cases, numerical features are scaled while categorical features are left unchanged.

Compute Minimum and Maximum Values: For each feature to be scaled, calculate the minimum and maximum values from the training dataset.

$$mini = min\{xi1, xi2, \ldots, xin\} \ldots\ldots\ldots\ldots\ldots (2)$$

pseudo code for Min-max Algorithm

The MinMaxScaler can be easily applied in Python as shown in the following code snippet:

```
From sklearn.preprocessing import
MinMaxScaler

numeric_cols =
df.select_dtypes(include=['float64',
'int64']).columns

scaler = MinMaxScaler()

df1[numeric_cols] =
scaler.fit_transform(df[numeric_cols])
```

Feature Selection

Feature selection is essential to reduce dimensionality and improve model performance. Here, two key techniques are applied: mutual information and Recursive Feature Elimination (RFE). Feature selection

Mutual Information

Mutual Information is a feature selection method that assesses the strength of dependency between two variables, calculating the extent to which knowledge of one variable reduces uncertainty about another (GeeksforGeeks, 2024). This technique was applied, and the top 10 features with the highest mutual information scores were selected, as these were considered the most informative for distinguishing between attack and normal traffic. The general equation for MI is given

DOI: 10.56892/bima.v8i4.1156

below. Figure 2 show the sample of feature selected using Mutual Information Technique

$(X; Y)$

$$= \sum_{x \in X} \sum_{y=Y} p(x,y)\ log \begin{pmatrix} p(x,y) \\ p(x)p(y) \end{pmatrix} \ldots . (3)$$

Hence,

- **I(X;Y)**: The **Mutual Information (MI)** between feature set X (the features selected from the dataset) and the target variable Y (the label indicating whether the traffic is benign or a DDS attack)
- **X**: The set of **input features** from the CICDDS2019 dataset, such as SourcePort, DestinationPort, FlowDuration, TotalFwdPackets, etc. (the attributes of the network traffic)
- **Y**: The **target variable**, which is the label or classification indicating the outcome (0 for benign traffic, 1 for DDoS attack).
- **p(x,y)**: The **joint probability distribution** of X (features) and Y (the target label).
- **p(x)**: The **marginal probability distribution** of the feature X
- **p(y)**: The **marginal probability distribution** of the target Y

```
MinPacketLength          0.256429
FwdPacketLengthMax       0.256363
PacketLengthMean         0.256205
AvgFwdSegmentSize        0.256177
AveragePacketSize        0.255805
MaxPacketLength          0.255751
FwdPacketLengthMin       0.255627
FwdPacketLengthMean      0.255516
SubflowFwdBytes          0.255496
TotalLengthofFwdPackets  0.255301
dtype: float64
```

Figure 2: Sample of feature selected using Mutual Information Technique.

Recursive Feature Elimination with Cross-Validation (RFECV)

RFECV, as a feature selection technique, chooses the most pertinent features that will allow good prediction by the model. For our dataset, the number of cross-validations used for RFECV was set to 4. This technique is beneficial in that it uses cross validation together with the process of feature of elimination. It was applied to iteratively remove less significant features to optimize the model's performance. This technique not only identifies the most predictive features but also ensures that the selected features contribute to the overall stability and accuracy of the model. The RFECV process resulted in 13 features being selected.

```
1 selected_features_rfecv, selected_features_rfecv.shape
```

```
(Index(['FwdPacketLengthMax', 'FwdPacketLengthMin', 'FwdPacketLengthMean',
        'FlowBytes/s', 'FwdIATTotal', 'FwdIATMax', 'FwdIATMin',
        'MinPacketLength', 'MaxPacketLength', 'PacketLengthMean',
        'AveragePacketSize', 'AvgFwdSegmentSize', 'SubflowFwdBytes'],
       dtype='object'),
 (13,))
```

Figure 3: Sample of 13 Selected Features by RFECV Technique

The model was trained using the intersection of the features selected by both MI and RFECV. The distribution of the dataset was in the ratio 70:30, ensuring a balanced distribution of normal and attack instances. This split provides a robust basis for training the model and evaluating its performance on unseen data. The dataset is divided into two parts: a training set and a testing set. In this study, 70% of the data (7,000 samples) is used for training, while 30% (3,000 samples) is used for testing. The splitting is done to ensure that the model learns on one portion of the data and is evaluated on another.

DOI: 10.56892/bima.v8i4.1156

Table 1: shows the data splitting for training.

Dataset	Number of Sample	Percentage(%)
Training Set	7 000	70
Testing Set	3 000	30
Total	10 000	100

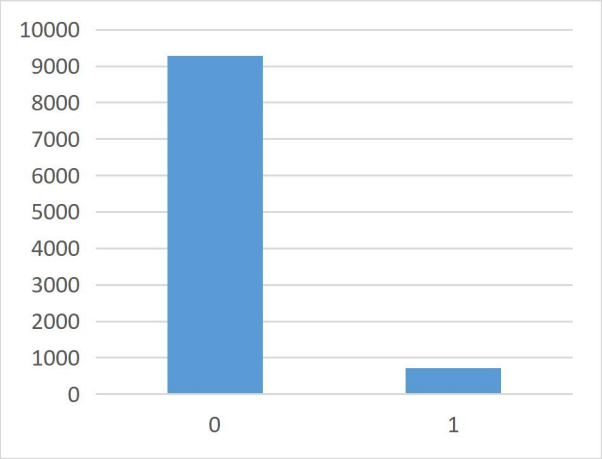

Figure 4: Dataset Distribution.

Keys:
0 = Non-malicious attack
1 = Malicious attack

Classification

SVM-AdaBoost Hybrid Model for Non Real-Time DDoS Attack Detection

This study presents a hybrid approach that combines Adaptive Boosting (AdaBoost) and Support Vector Machines (SVM) to improve the detection of DDoS attacks in IoT networks. This hybrid model relies heavily on Support Vector Machines (SVM) because they are very good at binary classification tasks, including detecting DDoS attacks. SVM maximises the margin between these two classes by identifying the best hyperplane to divide benign traffic from attack traffic. This makes SVM a good fit for detecting attack patterns in the context of DDoS detection based on network traffic parameters such as packet sizes, flow rates, and byte counts (Cortes & Vapnik, 1995). SVM, however, has trouble with noisy or unbalanced data, which is typical in Internet of Things networks.

This is where the SVM component is enhanced using AdaBoost.The model incorporates AdaBoost in order to overcome these constraints. AdaBoost is a boosting method that concentrates on incorrectly classified examples to enhance the performance of weak classifiers. AdaBoost iteratively trains SVM classifiers in the hybrid SVM-AdaBoost model, giving instances that were incorrectly categorised in prior rounds larger weights.

Consequently, this procedure produces a number of SVM classifiers that, when integrated, yield a model with higher accuracy. AdaBoost's capacity to improve the model is particularly helpful in identifying DDoS attacks that are subtle or low-frequency, which an SVM alone could overlook. AdaBoost and SVM together offer a number of benefits for non-real-time DDoS detection. The code snippet below demonstrates hyperparameter tuning for AdaBoost-SVM model for classification

```
from sklearn.model_selection import GridSearch CV from sklearn.ensemble import AdaBoostClassifier

from sklearn.svm import SVC

svm = SVC(probability=True)

adaboost = AdaBoostClassifier(base_estimator=svm)

param_grid = {

  'base_estimator__C': [0.1, 1, 10],

'n_estimators': [50, 100, 150],

}

grid_search= GridSearchCV(estimator=adaboost, param_grid=param_grid, cv=5, n_jobs=-1)
```

DOI: 10.56892/bima.v8i4.1156

```
grid_search.fit(X_train, y_train)
best_params = grid_search.best_params_
best_score = grid_search.best_score_
print(best_params)
print(best_score)
```

Algorithm: SVM-AdaBoost (Pseudocode)

$Procedure$:

$For\ t = 1\ to\ T$

$Train\ a\ base\ SVM\ classifier\ ht\ (x) using\ the$

$weighted\ dataset\ D\_t$

$Solve\ the\ SVM\ optimization\ problem\ with\ D_t\ as\ the\ sample\ weights$

$Compute\ the\ weighted\ classification\ error$:

$Et = \sum\big(D = t\ (i) * I\ (ht\ (xi) \neq yi)\big), where$

$I\ (condition) = 1\ if\ the\ condition\ is\ true,$

$Compute\ the\ weight\ of\ the\ classifier$:

$at = 0.5 * \log\ (1 - et)/et)$

$Update\ the\ sample\ weights$:

$Normalize\ D_t + 1\ so\ that \sum D_t + 1(i) = 1$

$Combine\ the\ classifiers\ into\ the\ final\ hypothesis$:

$H\ (x) =\ sign\ (\sum\big(at * ht(x)\big) for\ t = 1, \ldots., T$

$Return$:

$H = (x)$

Performance Evaluation Metrics

The model's performance was evaluated using several key metrics including Accuracy, Precision, Recall, F1 score, ROC Curve and AUC.

Accuracy:

The percentage of correctly categorised instances (both benign and attack traffic) relative to the total number of instances is known as accuracy, and it is a crucial parameter. Accuracy serves as a preliminary indicator of the overall validity of the model for both SVM and AdaBoost.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad \ldots\ldots\ldots\ldots\ldots (4)$$

Where:

TP is the number of true positives.

TN is the number of true negatives.

FP is the number of false positives.

FN is the number of false negatives.

Precision

Precision measures the proportion of true positive predictions among all positive predictions made by the model; it is crucial

DOI: 10.56892/bima.v8i4.1156

in scenarios where the cost of false positives is high.

$$\text{Precision}=\frac{TP}{TP+FP} \quad \text{.....(5)}$$

Recall (also known as Sensitivity or True Positive Rate) measures the proportion of true positive predictions among all actual positive instances; recall is particularly important in contexts where missing an attack (false negative) could have severe consequences.

$$\text{Recall}=\frac{TP}{TP+FN} \quad \text{.....(6)}$$

Score

The F1 Score is the harmonic mean of precision and recall and provides a single metric that balances both; it is especially useful when dealing with imbalanced datasets, where one class (e.g., normal traffic) is more prevalent than the other (e.g., attack traffic)

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad \text{.....(7)}$$

ROC-AUC

The Receiver Operating Characteristic (ROC) Curve plots the true positive rate (recall) against the false positive rate (FPR), which is defined as:

$$\text{FPR}=\frac{FP}{FP+TN} \quad \text{.....(8)}$$

The Area Under the ROC Curve (AUC-ROC) is a single scalar value that summarizes the performance of the model across all classification thresholds. AUC ranges from 0 to 1, with a value closer to 1 indicating a model with excellent discriminatory ability

RESULTS AND DISCUSSION

Result of Analysis of feature selection using Mutual Information (MI) and Recursive Feature Elimination with Cross-Validation (RFECV)

The dataset are labeled accordingly. To assess how well the trained model performed, the dataset was divided into training and testing sets during the feature selection processes. Once the model was trained and validated, the system performance was evaluated on a separate test set, which was not used during training. Table 2 provides a detailed breakdown of the model's performance in terms of training time and testing time on both Mutual Information (MI) and Recursive Feature Elimination with Cross-Validation (RFECV) using CICIDS2019 dataset

Table 2: Observed training and testing time of Mutual Information (MI) and Recursive Feature Elimination with Cross-Validation (RFECV) using CICDDS2019 dataset.

Feature Selection Algorithm	Training Time	Testing Time
Mutual Information	1.4320sec	1.5713sec
Recursive Feature Elimination with Cross-Validation	1.2031sec	1.3722sec

Table 3 presents the discussion of the training time and testing time of Mutual Information (MI) and Recursive Feature Elimination with Cross-Validation (RFECV) using CICIDS2019 dataset. However, the experimental results from the two feature selection were compared in Table 2 which shows that Recursive Feature Elimination with Cross-Validation (RFECV) outperform Mutual Information (MI) on CICIDOS2019 dataset.

Result SVM-AdaBoost Classifier on DDoS attack detection system on CICIDS2019 dataset

To investigate the robustness of the SVM-AdaBoost on DDoS attack detection, this work presents the following experiments on CICIDS2019 datasets. For each of the performance metrics listed in Tables 3, the

DOI: 10.56892/bima.v8i4.1156

SVM-AdaBoost of the DDoS detector was used to classify the training feature set and the DDoS was determined using the Accuracy, Precision, Recall, F1 score, AUC-ROC Tables 3 shows the results of the SVM-AdaBoost Classifier on CICIDS2019 dataset for DDoS detection.

Table 3: Result of SVM-AdaBoost Classifier on CICIDS2019 dataset for DDoS detection.

Metrics	SVM-AdaBoost
Accuracy	99.9%
Precision	99.5%
Recall	99.1%
F1 score	99.3%
AUC-ROC	100%

From Table 3, the SVM-AdaBoost classifier achieved an accuracy of approximately 99.9.%, the precision is 99.5%, Rcall 99.1%, F1score 99.3% and AUC-ROC 100%.

Comparison of SVM-Adaboost with Other Studies

The study conducted a comparative analysis of the develop model against established approaches in the field, and the findings indicated that the developed model delivered results that are on par with the best-performing models in the literature. Figure 5 display the Comparative Result Analysis (SVM-AdaBoost and SVM)

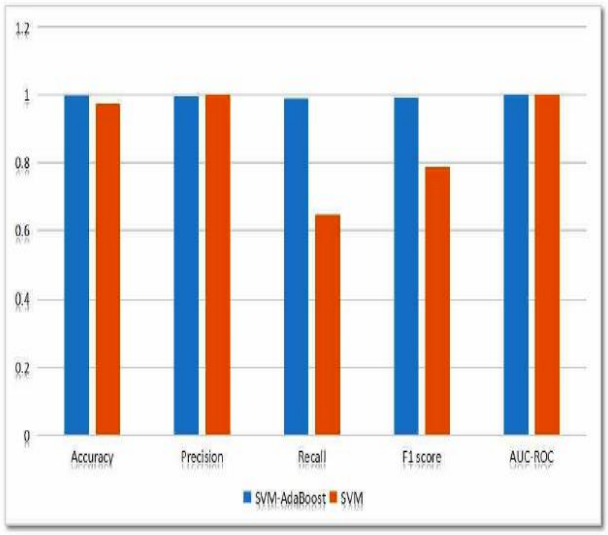

Figure 5: Comparative Result Analysis (SVM-AdaBoost vs. SVM).

From Figure 5, the comparison between the Support Vector Machine (SVM) classifier and the SVM-AdaBoost classifier reveals significant differences in their performance across key evaluation metrics, underscoring the advantages of ensemble methods in enhancing detection capabilities.

Table 4: Comparison of SVM-AdaBoost with Other Studies.

Research Paper	Author and Year	Methodology	Result
Dynamic analysis of malwarae intrusion in Mobile devices Using AdaBoost Algorithm, KNN AND SVM Base Classifiers	Oyong, Ekong, Obot (2023)	Combination of KNN and SVM	Accuracy rates of 91.43%
examined Development of a Distributed Denial of Service Detection Model Using Ensemble Machine Learning Techniques	Abolarinwa et al., (2024)	The user interface was developed using HTML, CSS, and JavaScript frameworks.	The Bagging Ensemble approach outperformed the other models, with approximately 99.5% accuracy, 97% precision, 94.88% recall, and 95.61% F1-score.
Proposed Model	2024	Esemble voting for SVM, NB, DT, LR & ANN	Ensemble voting Accuracy = 99.65, Precision=99%, Recall=100%, F1-Score=100%

DOI: 10.56892/bima.v8i4.1156

CONCLUSION

The feature selection process was critical in managing the high dimensionality of the data. Mutual Information and RFECV, when used together, proved to be highly effective in identifying a set of features that maximized the model's performance. The intersection of features selected by both techniques yielded 9 highly informative features, which provided a balance between model complexity and performance.

The integration of SVM as a classifier with AdaBoost significantly improved the robustness and accuracy of the intrusion detection model. The AdaBoost algorithm, by focusing on difficult-to-classify instances and iteratively refining the model, enhanced the detection capability, particularly for minority attack classes, gotten from the selected feature, the SVM-AdaBoost algorithm achieved an accuracy of 99.9%, precision of 99.5%, recall of 99.1%, and an F1 score of 99.3%. The ROC curve analysis further confirmed the model's effectiveness,

In conclusion, this research successfully demonstrated that the combination of Support Vector Machines with the AdaBoost ensemble method provides a robust and effective approach for detecting DDoS attacks in IoT networks. The model developed in this study showed high accuracy and generalization capabilities, effectively addressing the challenges posed by high-dimensional data and the need for efficient feature selection.

Recommendation

1. Further research is recommended to explore real-time implementations and the use of additional machine learning techniques to continue advancing the field of intrusion detection in IoT environments.

2. Future research and practical implementations should prioritize feature selection to enhance model performance and reduce computational overhead, particularly in resource-constrained IoT environments.

REFERENCES

Ashton, K. (2023). That 'Internet of Things' Thing. *Journal Information Technology*. Applications and Research Challenges.

Abolarinwa M. O., Adegoke, E. A., OJO O. E., Adewuya A. M., Bakare O. S. & Adigun O. I. (2024) Development of a Distributed Denial of Service Detection Model Using Ensemble Machine Learning Techniques. *Adeleke University Journal of Science (AUJS)*.https://aujs.adelekeuniversity.edu.ng.

Alamgir H. & Saiful I.(2024) Enhancing DDoS attack detection with hybrid feature selection and ensemble-based classifier: *Journal of Institute of Information and Communication Technology (IICT*), Bangladesh University of Engineering and Technology (BUET).

Arora, A., Yadav, S. K., & Sharma, K. (2018). Denial-of-Service (DOS) attack and Botnet. In Advances in information security, privacy, and ethics book series. https://doi.org/10.4018/978-1-5225-4100-4.ch008.

Alahmadi, D., Bamasag, O., Ullah, H., & Asghar, M. Z. (2023). Efficient Detection of DDoS Attacks Using a Hybrid Deep Learning Model with Improved Feature Selection. *Journal of Applied Sciences*, 11(24), 11634. https://doi.org/10.3390/app112411634.

Awad, M., & Fraihat, S. (2023). Recursive Feature Elimination with Cross-Validation with Decision Tree. Feature Selection Method for Machine

DOI: 10.56892/bima.v8i4.1156

Learning-Based Intrusion Detection Systems. *Journal of Sensor and Actuator Networks,* 12(5), 67. https://doi.org/10.3390/jsan12050067.

Dalal, S., Lilhore, U. K., Faujdar, N., Simaiya, S., Ayadi, M., Almujally, N. A. & Ksibi, A. (2023). Next-generation cyber attack prediction for IoT systems: leveraging multi-class SVM and optimized CHAID decision tree. *Journal of Cloud Computing Advances Systems and Applications*, 12(1). https://doi.org/10.1186/s13677-023-00517-4.

Dasari, K. B. & Devarakonda, N. (2021). Detection of different DDOS attacks using machine learning classification algorithms. *Journal of Sensor and Actuator Networks*, 12(5), 67. https://doi.org/10.3390/jsan12050067.

Jallad, K. A., Aljnidi, M. & Desouki, M. S. (2020). Anomaly detection optimization using big data and deep learning to reduce false-positive. *Journal of Big Data*, 7(1). https://doi.org/10.1186/s40537-020-00346-Mendy, Z., & Elliot, S. (2023). Customizing SVM as a base learner with AdaBoost ensemble to learn from multi-class problems: A hybrid approach AdaBoost-MSVM. Knowledge-Based Systems, https://doi.org/10.1016/j.knosys.2021.106845.

Mahrukh, R., Muhammad, S., Ayesha, A., Shazia, A., Faiza, I., Ángel, K. C. and Imran A. (2023) Distributed Denial of Service Attack Detection in Network Traffic Using Deep Learning Algorithm. *Journal of AdvanceTechnology* https://www.mdpi.com/journal/sensors.

Najafimehr, M., Zarifzadeh, S. & Mostafavi, S. (2022). A hybrid machine learning approach for detecting unprecedented DDoS attacks. *The Journal of Supercomputing*, 78(6), 8106–8136. https://doi.org/10.1007/s11227-021-04253-x.

Natesan, P., Balasubramanie, P. & Gowrison, G. (2012). Improving the Attack Detection Rate in Network Intrusion Detection using the Adaboost Algorithm. *Journal of Computer Science*, 8(7), 1041–1048. https://doi.org/10.3844/jcssp.2012.1041.104Oyong, S. B., Ekong, U. O. & Obot, O. U. (2023) Dynamic analysis of malwarae intrusion in mobile devices using adaboost algorithm, KNN AND SVM base classifiers. *World Journal of Applied Science and Technology,* https://dx.doi.org/10.4314/WOJAST.v15i1.78 Vol. 15 No. 1 (2023) .78 – 84.

Richard, K. A. & Micheal, M. S. (2023). Anomaly detection optimization using big data and deep learning to reduce false-positive. *Journal of Big Data*, 7(1). https://doi.org/10.1186/s40537-020-00346-1.

Sambangi, S., & Gondi, L. (2020). A Machine Learning Approach for DDoS (Distributed Denial of Service) Attack Detection Using Multiple Linear Regression. MDPI. *Journal of Kufa for Mathematics and Computer*, 10(1), 102-107 https://doi.org/10.3390/proceedings2020063051.

Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H. & Deng, S. H. (2023). Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology*, 17(1), 26–40. https://doi.org/10.11989/jest.1674-862x.80904120.

Yaras, S. & Dener, M. (2024). IoT-Based Intrusion Detection System using new

DOI: 10.56892/bima.v8i4.1156

hybrid deep learning algorithm. *Journal of Electronics Engineering, 13(6), 1053*. https://doi.org/10.3390/electronics13061053.

Yakub, K. S., Aremu, I. A., Sanjay, M., Monica K. H. & Ricardo, C.P. (2022) A machine learning-based intrusion detection for detecting internet of things network attacks. *Journal*. of *Alexandria Engineering* Published by Elsevier on behalf of Faculty of Engineering, Alexandria University.

Zhang, Z., Hu, Y., & Ji, Y. (2023). Normalization techniques in deep learning. *Journal of Computational and Applied Mathematics*, 368, 112396.