DDOS ATTACKS DETECTION USING NAIVE BAYES CLASSIFIER

By

GUDIPUDI DAYANANDAM * SRINIVASA REDDY E. ** BUJJI BABU D. ***

* *Department of Computer Science, Government Degree College, Kodur(RS), Annammayya, Andhra Pradesh, India.*
** *Department of Computer Science and Engineering, University College of Engineering and Technology, Guntur, Andhra Pradesh, India.*
*** *Department of Computer Science and Engineering, QIS College of Engineering and Technology, Ongole, Andhra Pradesh, India.*

https://doi.org/10.26634/jit.13.3.21109

Date Received: 15/08/2024 *Date Revised: 27/08/2024* *Date Accepted: 06/09/2024*

ABSTRACT

Internet usage has become essential for effective and timely communication, e-commerce activities, and financial transactions, contributing to a more sophisticated lifestyle. However, these activities are increasingly vulnerable to internet threats and fraud. A Distributed Denial of Service (DDoS) attack is a prevalent internet threat that disrupts the normal traffic of a victim server by overwhelming the target infrastructure with a flood of internet traffic. The primary aim of attackers is to create uncertainty for individuals or organizations, typically seeking financial gain or aiming to damage an organization's reputation. Notably, during the Russia-Ukraine war, significant DDoS attacks targeted Ukrainian bank servers to disrupt financial services for customers. This study employs the Naive Bayes model with 10-fold cross-validation to detect DDoS attacks. Naive Bayes, a widely recognized machine learning algorithm, demonstrated superior performance. The results revealed an average accuracy of 0.999, outperforming existing machine learning-based DDoS attack detection techniques.

Keywords: DDoS Attacks, Caret Package, Naïve Bayes, KDD'99 Dataset, Machine Learning Algorithms.

INTRODUCTION

Cybersecurity has become a major topic of discussion in recent times. This concern primarily arises from the enormous internet traffic generated by advancements in technology, as organizations and individuals increasingly rely on various web and mobile applications to achieve their commercial objectives. Cyberattacks have caused significant damage and financial losses (Radware, 2024). DDoS attacks fall under this category of cyberattacks. These attacks are based on the principle of DoS (Denial of Service) attacks (Onelogin, n.d.). While DoS attacks primarily target a single server, their main objective is to disrupt services for legitimate users.

Distributed Denial of Service (DDoS) attacks, unlike DoS attacks, focus on a many-to-one mapping concept, where a large number of attackers target a single server. The need for machine learning research is expected to grow 38% by 2026 (Chenniappanadar et al., 2022). The primary goal of a DDoS attack is to overwhelm a server with an excessive number of requests, exceeding its capacity. The attack persists until the server crashes or stops responding. If a remedy is delayed, the financial losses can be significant.

Detecting a single fake user is easy, as the server can deny messages from that particular user. This type of attack is called a Denial of Service (DoS) attack. However, if multiple fake requests are received from multiple users, it becomes very difficult to detect the sources and isolate those users. In such cases, it becomes impossible to provide services to authorized users. This type of attack is called a Distributed Denial of Service (DDoS) attack.

Reasons for DDoS Attacks

- *Ransom:* After performing DDoS attacks, the attackers demand ransom to decrypt their files.

- *Hacktivism:* Hacktivists use their voice to carry out DDoS attack to show their backing or resistance to a company or individual.
- *Competition:* Companies frequently accuse their competitors when faced with DDoS attacks. To damage the reputation of rivals, they may spend large amounts of money executing DDoS attacks on them.

Types of DDoS Attacks

DDoS attacks can be classified into three types (Testbytes, 2024). They are:

- Volumetric based attacks.
- Protocol based attacks.
- Application layer attacks.

Volumetric Based Attacks

Volumetric-based attacks are conducted by overwhelming a server with massive traffic, draining its bandwidth. A DNS amplification attack is an example of such volume-based attacks. In this scenario, hackers use spoofed IP addresses to send requests to a DNS server. The DNS server then sends a response to the target server. When the target server receives an overwhelming number of responses beyond its capacity, it crashes and becomes unable to respond to authorized users, as Figure 1 shows.

Protocol Based Attacks

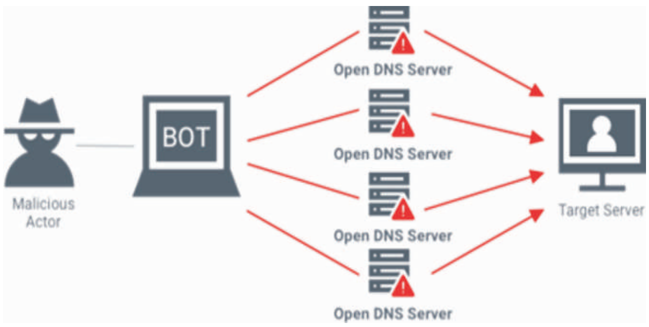
Protocol-based attacks aim to exhaust the resources of a server or networking systems such as firewalls, load balancers, or routing engines. A SYN flood attack is an example of such protocol-based attacks. Using the 3-way handshake protocol, a connection is established between two parties through the exchange of three messages: SYN, SYN-ACK, and ACK. In SYN flood attacks, the attacker floods the server with a large number of SYN packets, aided by IP spoofing. The server responds with SYN-ACK, but since the client's IP addresses are spoofed, the server never receives the ACK messages needed to complete the handshake. As a result, the server's buffer becomes filled with numerous incomplete requests and is unable to respond to legitimate client requests, as show in Figure 2. Smurf attacks are another example of protocol-based attacks.

Application Layer Attacks

The server generates a response to requests from any client at the application layer. For example, when a user enters a website address in their web browser, the client sends an HTTP request to the server for a particular web page. The server then responds with the requested information. This process of searching for, fetching, and sending information to the client occurs on the application server. Application layer attacks happen when an attacker repeatedly sends bogus requests for the same resource, ultimately flooding the server and rendering it unable to respond to legitimate clients.

HTTP flood attacks are one such type of attack in which attackers continuously send a large number of HTTP requests to the server using different IP addresses. As shown in Figure 3, when the server's buffer becomes full, it is unable to respond to legitimate client requests.

Machine learning algorithms are focused on predicting



Figuer 1. DNS Based DDoS Attacks

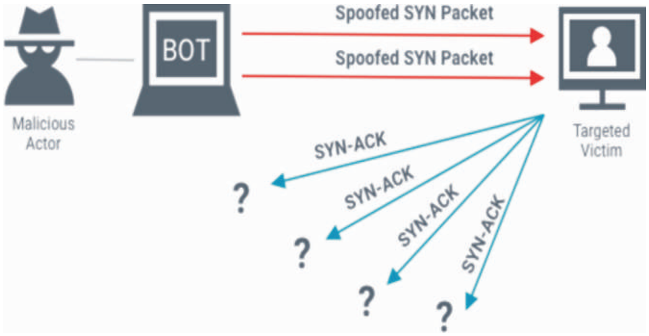


Figure 2. TCP SYN Flood Based DDoS Attacks

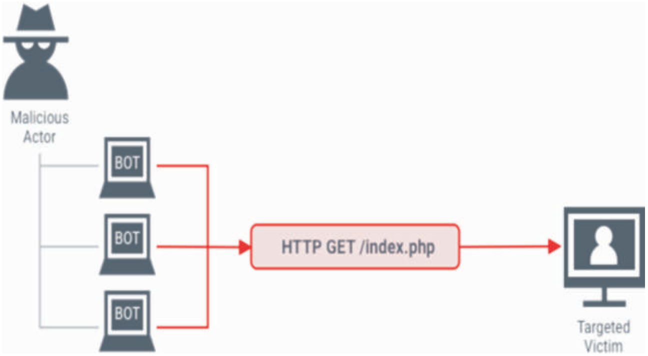


Figure 3. HTTP Flood Based DDoS Attacks

the future based on previous knowledge. Classification and Clustering techniques play a vital role in predicting future outcomes.

The specific objective of attack detection is to find the deviations or other anomalies against security rules. Anomaly detection can be done by using machine learning technique which enhances the effectiveness of algorithm and reduces the detection time. Organizations use IOT devices against these types of anomalies. There is a need for methods to defend against security attacks on IoT devices. Traditional IDS methods are typically too sophisticated to secure IoT devices (Saba et al., 2022).

1. Related Works

Biswas (2018) and Chenniappanadar et al. (2022) developed the SPLR model for intrusion detection, which enhances attack categorization. This method handles large datasets and addresses overfitting and feature redundancy by performing feature selection and classification simultaneously.

Li et al. (2019) proposed a PCA-based feature reduction and RNN-based prediction model. Their model, using the KDD dataset, showed that PCA-RNN provides better detection accuracy.

Gao et al. (2019) suggested deploying data mining techniques to analyze alarms received by IDS/IPS systems and deep defense network security architectures. Their approach rapidly identifies attacks, boasting high detection rates and low false positives.

Gurulakshmi and Nesarani (2018) discussed DoS attacks, which aim to prevent authorized users from accessing servers or using their legitimate information. In these attacks, an attacker overloads the server with excessive queries, causing the server to be unable to respond to legitimate requests.

Sarkar et al. (2020) identified tree-based strategies as the most effective machine learning classification methods for dataset creation.

Fischer (2005) proposed "IntruDTree," a method that evaluates the importance of security features before reinforcing a tree for attack detection.

Martinez Torres et al. (2012) explored various ways to apply cybersecurity from a legal perspective. They also investigated machine learning algorithms for detecting a range of attacks.

Shang (2024) used Naïve Bayes and Random Forest algorithms for detecting and mitigating DDoS attacks. Their findings showed that the Naïve Bayes method outperformed Random Forest in identifying false and actual transmission rates, achieving an accuracy of 97.3%.

Naiem et al. (2023) addressed several challenges related to the Gaussian Naïve Bayes classifier, such as the zero-frequency problem and feature independence assumptions. They proposed a framework that includes feature selection, data preprocessing, and algorithm enhancements. These improvements reportedly increased classifier accuracy by 2% and the overall average accuracy and precision by 1.5%.

JFM Garcia and Blandon (2022) developed an Intrusion Detection and Prevention System (IDPS) named Dique, leveraging deep learning to identify and mitigate DoS attacks. Their system, using a multi-layered Deep Feed Forward neural network trained on the CICDDoS2019 dataset, achieved an accuracy of 0.994.

Mandala et al. (2022) used the CICIDS2018 dataset and applied information gain for feature selection to identify the most influential features in detecting DDoS attacks. They employed the Naïve Bayes method to build a prediction model, improving detection accuracy from 65% without feature selection to 69.6% with feature selection.

Sudar et al. (2021) focused on identifying and mitigating

DDoS attacks within Software-Defined Networks (SDN) using machine learning methods. This research addresses the increasing concern of DDoS attacks in SDN environments, which are critical for modern network management due to their centralized control and programmability. The study explores various machine learning algorithms to detect anomalies in network traffic that indicate DDoS attacks.

Banerjee and Chakraborty (2021) discussed the use of machine learning algorithms to detect DDoS attacks in SDN environments. SDN is a network architecture that allows for intelligent, centralized control using software applications. The authors explored different machine learning techniques to enhance detection accuracy and response time to DDoS attacks, which pose significant threats to network security.

Usha et al. (2021) aimed to develop an efficient system for detecting and classifying DDoS attacks using various machine learning algorithms. The authors employed techniques such as Extreme Gradient Boosting (XGBoost), K-Nearest Neighbour (KNN), Stochastic Gradient Descent (SGD), Naive Bayes, and deep learning architectures like Convolutional Neural Networks (CNN). Among these, XGBoost achieved the highest accuracy in detecting and classifying DDoS attacks, with CNN and KNN also showing comparable performance.

AlMomin and Ibrahim (2020) explored various machine learning algorithms to improve the detection accuracy of DDoS attacks. By combining these algorithms, the authors aimed to create a robust detection system that can efficiently identify and respond to such attacks, ensuring network security and stability.

2. Naïve Bayes classifier

Naïve Bayes algorithm is based on Bayes theorem, which is used to solve the classification problem using probabilistic approach. In this approach, predictor variables are independent of each other. The outcome of a model depends on a set of independent variables.

2.1 Why Naïve Bayes is 'Naïve':

Naïve Bayes is called Naïve because it considers each predictor variable independent of each variable even though there is a correlation between the variables in real-world problems.

2.2 The Math Behind Naïve Bayes:

The Bayes' Rule forms the foundation of the Naïve Bayes classifier. It is used to calculate conditional probability, which represents the likelihood of an event occurring based on the occurrence of previous events.

Mathematically Bayes theorem is

$$P(X|Y) = \frac{P(Y|X)\,P(X)}{P(Y)}$$

Where P(X|Y) = Conditional Probability of event 'X' occurring, given that event 'Y'

P(X) = Probability of event 'X' occurs,

P(Y) = Probability of event 'Y' occurs,

P(Y|X) = Conditional Probability of event 'Y' occurring, given that event 'X'.

The terminology of Bayesian theorem are as follows.

X is also called Proposition, whereas Y is Evidence.

P(X|Y) = Posterior

P(X) = Prior probability of the Proposition

P(Y) = Prior probability of the Evidence,

P(Y|X) = Likelihood

So Bayes theorem can be rewritten as follows.

$$\text{Posterior} = \frac{\text{(Likelihood).(Prior probability of Proposition)}}{\text{(Prior probability of Evidence)}}$$

Bayes' theorem can be rewritten in terms of the hypothesis as given in the following formula.

$$P(H|E) = \frac{P(E|H)\,P(H)}{P(E)}$$

Where H is Hypothesis and E is Evidence.

P(E|H) = Likelihood,

P(E) = Evidence Prior probability,

P(H) = Hypothesis Prior probability,

P(H|E) = Posterior

This Bayes theorem is used for single predictor variable.

2.3 Naïve Bayes on KDD'99 Dataset:

- *Observations:* Predictor variables are dependent variables and output variables are class variables which contain the value 0 or 1. Output will be in form of two classes 1 and 0, where 1 indicates output variable contains 'smurf' attack and 0 indicates output variable doesn't contain 'smurf' attack. So, output variable is converted to categorical variable. The structure of the dataset is observed so that the dataset description is useful for finding missing values. Missing values in the dataset can impact the performance of the machine learning model. Therefore, it is essential to clean the dataset by handling these missing values. Next, data splicing is performed, where the dataset is divided into two parts.

- *Training Dataset:* It is used to build and train the machine learning model.
- *Testing Dataset:* It is used to evaluate the efficiency and performance of the model.

To store the value of response variable, separate variables are created which are useful for comparison of outcome of the training and testing phase. X and Y variables are used for predictor variable and response variable respectively. Next, a predictive model is created using the Naïve Bayes classifier. Finally, the model is evaluated using a confusion matrix. This evaluation is performed by inputting the testing dataset into the predictive model to assess its efficiency.

3. Methodology

This section focuses on the methodology of the proposed method. The KDD'99 dataset, which is a benchmark for IDS, was used. The KDD'99 dataset contains 43 features, and the label was selected as the output variable, which includes the 'smurf' attack. The 'smurf attack' is one of the Distributed Denial of Service (DDoS) attacks.

The proposed methodology, as shown in Figure 4, was followed throughout the experiment.

3.1 Step by Step process of Naïve Bayes:

- *Step 1:* Install and load the required packages. For this experiment, the 'caret' package was used.
- *Step 2:* Import the dataset.

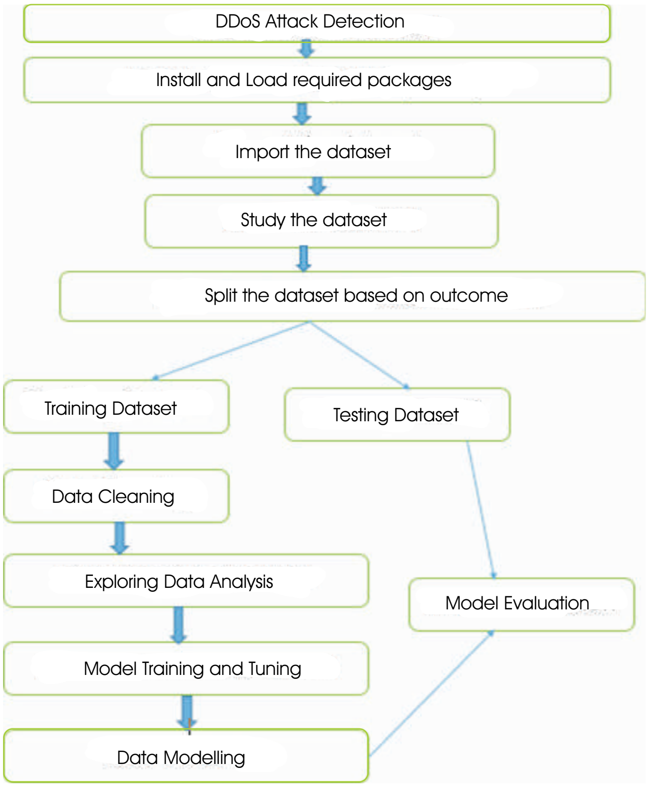


Figure 4. Flow Diagram

- *Step 3:* Studying the dataset. i.e. identifying dependent and independent variable.
- *Step 4:* Data cleaning and Data analysis.
- *Step 5:* Model Training and Tuning.
- *Step 6:* Data Modelling.
- *Step 7:* Model Evaluation.

In Step 1, the 'caret' AME (2024) package was installed and loaded, containing the train() function to set up a grid of tuning parameters for various classification and regression problems. In Step 2, the KDD'99 dataset was read using the import function. Step 3 involved identifying input and output variables, where the input variables are independent and the output variable is the dependent variable. The dataset was then split into training and testing datasets based on the outcome. The outcome was converted into a factor variable instead of a data frame before performing the cleaning process. In Step 4, the cleaning process was applied to find any missing values or outliers, which could affect the performance of the model. More missing values or outliers would lead to a decrease in performance. The relationship between different variables was examined through exploratory analysis. Step 5 involved using the trainControl() function for model training and tuning. In Step 6, the Naïve Bayes model was applied using the train() function. The parameters of the train() function included the training data, a vector containing the outcome, and the model name ('nb') indicating classification with the Naïve Bayes model. The trainControl() argument instructed the trainer to use 10-fold cross-validation. The training dataset was randomly divided into 10 equal-sized sub-samples. Of these 10 sub-samples, 9 were used for training data and 1 for validation data. This cross-validation process was repeated 10 times, resulting in 10 estimates, which were averaged to produce a single estimate. The output showed a kappa value of 0.998, indicating excellent prediction performance. In Step 7, the proposed model was used for prediction.

4. Results and Discussion

The model is evaluated and compared with various existing detection algorithms using the KDD'99 dataset (UCI KDD, 1999). The dataset is divided into a training dataset and a testing dataset based on outcomes. The evaluation is performed using a confusion matrix, which is a square matrix shown in Table 1. TP, FP, FN, and TN are defined as:

- *True Positive:* The predicted value is positive, and the actual value is also positive.
- *False Positive:* The predicted value is positive, but the actual value is negative.
- *False Negative:* The predicted value is negative, but the actual value is positive.
- *True Negative:* The predicted value is negative, and the actual value is also negative.

Performance of the model can be calculated by using following formulas.

N=Total Predictions	Actual: Yes	Actual: No
Predicted: Yes	True Positive (TP)	False Positive (FP)
Predicted: No	False Negative (FN)	True Negative (TN)

Table 1. Confusion Matrix

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$$

Accuracy provides the ratio of correctly identified attacks with the overall traffic i.e. which contains attacks and normal traffic.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Precision will give us fraction of correct predictions.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Sensitivity is the fraction of DDoS attacks that are correctly predicted.

$$\text{F1 Score} = \frac{2 * \text{recall} * \text{precision}}{recall + precision}$$

F1 Score is the weighted average of Recall and Precision.

$$\text{False Alarm Rate} = \frac{\text{FP}}{FP + TN}$$

Where TP means True Positive, FP means False Positive, FN means False Negative and TN means True Negative.

The proposed method is compared with other existing DDoS attack detection methods.

Table 2 shows the comparison between existing approaches and the newly implemented Naïve Bayes method with cross-validation. The analysis demonstrates that the newly applied approach outperforms others across various parameters, including accuracy, sensitivity, precision, and F1 score, as shown in Figure 5, which presents the accuracy graph for the different models.

Conclusion

A DDoS attack disrupts the services of authorized users and has grown exponentially in recent times, particularly

S.No	Algorithm	Accuracy	Sensitivity	Precision	F1 Score
1	PCA-RNN [11]	0.9872	0.9872	0.9810	0.9810
2	SPLR [2]	0.986	0.986	0.9866	0.9866
3	Y Shang[17] used Naïve Bayes	0.973	0.972	0.97	0.972
4	Sarah N et al. [18]	0.98	0.982	0.983	0.98
5	Naïve Bayes with Cross Validation	0.999	0.998	0.996	0.998

Table 2. Comparison of Proposed Model with Existing Models

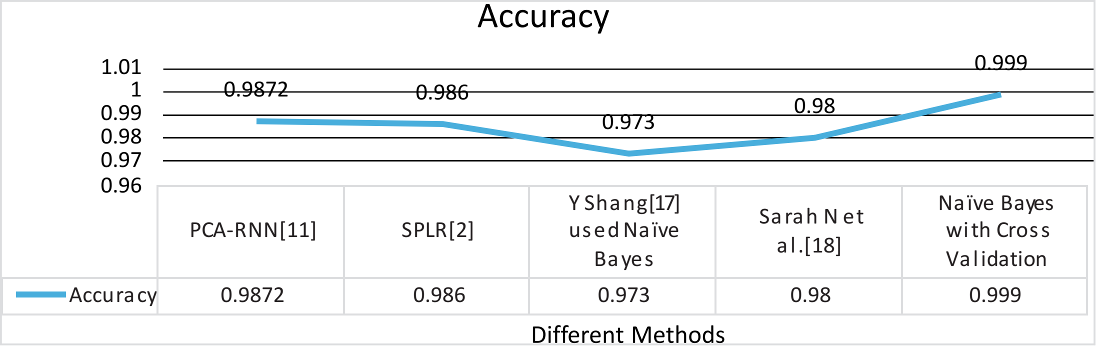


Figure 5. Accuracy Graph for the Different Models

during the Russia-Ukraine war. In these DDoS attacks, hackers took control of bank servers, rendering services unavailable to customers. Therefore, detecting DDoS attacks has become a prominent area of focus.

In this paper, smurf attacks and normal traffic in the KDD'99 dataset were classified using the Naive Bayes classifier. Choosing the best algorithm for machine learning is crucial as it impacts the model's performance. The caret package in the R language facilitated improved results. Functionally, the caret package enabled splitting the data based on the outcome variable, resulting in better outcomes compared to other methods.

In future work, the technique will be applied to more real-time datasets to study abnormal conditions of network traffic in different application domains.

References

[1]. AlMomin, H., & Ibrahim, A. A. (2020, June). Detection of distributed denial of service attacks through a combination of machine learning algorithms over software defined network environment. In *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (pp. 1-4). IEEE.

https://doi.org/10.1109/HORA49412.2020.9152873

[2]. AME. (2024). *Annals of Translational Medicine.* Retrieved from

https://atm.amegroups.org/

[3]. Banerjee, S., & Chakraborty, P. S. (2021, February). To detect the distributed denial-of-service attacks in SDN using machine learning algorithms. In *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)* (pp. 966-971). IEEE.

https://doi.org/10.1109/ICCCIS51004.2021.9397068

[4]. Biswas, S. K. (2018). Intrusion detection using machine learning: A comparison study. *International Journal of Pure and Applied Mathematics,* 118(19), 101-114.

[5]. Chenniappanadar, S. K., Gnanamurthy, S., Sakthivelu, V. K., & Kaliappan, V. K. (2022). A supervised machine learning based intrusion detection model for detecting cyber-attacks against computer system. *International Journal of Communication Networks and Information Security,* 14(3), 16-25.

[6]. Edureka. (2024). *Trending Courses.* Retrieved from

https://www.edureka.co/

[7]. Fischer, E. A. (2005, February). *Creating a National Framework for Cybersecurity: An Analysis of Issues and Options.* Congressional Information Service, Library of Congress.

[8]. Gao, Y., Wu, H., Song, B., Jin, Y., Luo, X., & Zeng, X. (2019). A distributed network intrusion detection system for distributed denial of service attacks in vehicular ad hoc network. *IEEE Access,* 7, 154560-154571.

https://doi.org/10.1109/ACCESS.2019.2948382

[9]. Garcia, J. F. C., & Blandon, G. E. T. (2022). A deep learning-based intrusion detection and preventation system for detecting and preventing denial-of-service attacks. *IEEE Access,* 10, 83043-83060.

https://doi.org/10.1109/ACCESS.2022.3196642

[10]. **Gurulakshmi, K., & Nesarani, A. (2018, May).** Analysis of IoT bots against DDOS attack using machine learning algorithm. In *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 1052-1057). IEEE.

https://doi.org/10.1109/ICOEI.2018.8553896

[11]. **Li, Q., Meng, L., Zhang, Y., & Yan, J. (2019).** DDoS attacks detection using machine learning algorithms. In *Digital TV and Multimedia Communication: 15th International Forum, IFTC 2018, Shanghai, China, September 20–21, 2018, Revised Selected Papers 15* (pp. 205-216). Springer Singapore.

https://doi.org/10.1007/978-981-13-8138-6_17

[12]. **Mandala, S., Ramadhan, A. I., Rosalinda, M., Yafooz, W. M., & Khohar, R. H. (2022, December).** Ddos detection by using information gain-naïve bayes. In *2022 2nd International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)* (pp. 283-288). IEEE.

https://doi.org/10.1109/ICICyTA57421.2022.10038054

[13]. **Martínez Torres, J., Iglesias Comesaña, C., & García-Nieto, P. J. (2019).** Machine learning techniques applied to cybersecurity. *International Journal of Machine Learning and Cybernetics,* 10(10), 2823-2836.

https://doi.org/10.1007/s13042-018-00906-1

[14]. **Mukherjee, S., & Sharma, N. (2012).** Intrusion detection using naive Bayes classifier with feature reduction. *Procedia Technology,* 4, 119-128.

https://doi.org/10.1016/j.protcy.2012.05.017

[15]. **Naiem, S., Khedr, A. E., Marie, M., & Idrees, A. M. (2023).** Enhancing the efficiency of gaussian naïve bayes machine learning classifier in the detection of DDOS in cloud computing. *IEEE Access,* 11, 124597 - 124608.

https://doi.org/10.1109/ACCESS.2023.3328951

[16]. **Onelogin. (n.d.).** *What is a DDoS Attack?* Retrieved from

https://www.onelogin.com/learn/ddos-attack

[17]. **Radware. (2024).** *2021 – 2022 Global Threat Analysis Report.* Retrieved from

https://www.radware.com/2021-2022-global-threat-analysis-report/

[18]. **Saba, T., Rehman, A., Sadad, T., Kolivand, H., & Bahaj, S. A. (2022).** Anomaly-based intrusion detection system for IoT networks through deep learning model. *Computers and Electrical Engineering,* 99, 107810.

https://doi.org/10.1016/j.compeleceng.2022.107810

[19]. **Sarker, I. H., Abushark, Y. B., Alsolami, F., & Khan, A. I. (2020).** Intrudtree: A machine learning based cyber security intrusion detection model. *Symmetry,* 12(5), 754.

https://doi.org/10.3390/sym12050754

[20]. **Shang, Y. (2024).** Prevention and detection of DDOS attack in virtual cloud computing environment using Naive Bayes algorithm of machine learning. *Measurement: Sensors,* 31, 100991.

https://doi.org/10.1016/j.measen.2023.100991

[21]. **Sudar, K. M., Beulah, M., Deepalakshmi, P., Nagaraj, P., & Chinnasamy, P. (2021, January).** Detection of distributed denial of service attacks in SDN using Machine learning techniques. In *2021 international conference on Computer Communication and Informatics (ICCCI)* (pp. 1-5). IEEE.

https://doi.org/10.1109/ICCCI50826.2021.9402517

[22]. **Testbytes. (2024).** *Testimonials.* Retrieved from

https://www.testbytes.net/

[23]. **UCI KDD. (1999).** *KDD Cup 1999 Data.* Retrieved from

https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[24]. **Usha, G., Narang, M., & Kumar, A. (2021).** Detection and classification of distributed DoS attacks using machine learning. In *Computer Networks and Inventive Communication Technologies: Proceedings of Third ICCNCT 2020* (pp. 985-1000). Springer Singapore.

https://doi.org/10.1007/978-981-15-9647-6_78

ABOUT THE AUTHORS

G. Dayanandam is currently working as a Lecturer in the Department of Computer Science at Government Degree College, Kodur (RS), Annammayya, Andhra Pradesh, India. He completed his Master of Technology (M.Tech) in Information Technology (IT) from Andhra University College of Engineering, Visakhapatnam, Andhra Pradesh, India, and is pursuing a Doctor of Philosophy (Ph.D.) in Computer Science and Engineering (CSE) from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. He has published more than 10 research papers in reputed peer-reviewed journals. He has also attended and presented research papers at various national and international conferences, with the proceedings indexed in Institute of Electrical and Electronics Engineers (IEEE) and Springer Link series. He has guided several undergraduate (UG) and postgraduate (PG) projects. His areas of interest include Network Security, Cryptography, and Machine Learning.

Dr. E. Srinivasa Reddy is a Professor in the Department of Computer Science and Engineering at University College of Engineering and Technology, Guntur, Andhra Pradesh, India, and the Principal at Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. He received his Master of Technology (M.Tech) from Visveswaraya Technological University, Karnataka, India, and his Doctor of Philosophy (Ph.D.) in Computer Science and Engineering from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. With 32 years of experience in teaching and research, he has published more than 100 research papers in national and international journals. His research interests include Digital Image Processing and Pattern Recognition.

Dr. D. Bujji Babu is currently a Professor in the Department of Computer Science and Engineering at QIS College of Engineering and Technology, Ongole, Andhra Pradesh, India. He completed his Master of Technology (M.Tech) in Computer Science and Engineering (CSE) from Jawaharlal Nehru Technological University, Kakinada, Andhra Pradesh, India, and his Doctor of Philosophy (Ph.D.) in Computer Science and Engineering from Acharya Nagarjuna University. He is a recognized research supervisor under Jawaharlal Nehru Technological University, Kakinada, Andhra Pradesh, and has guided several undergraduate (UG) and postgraduate (PG) projects. He is currently supervising four research scholars under Jawaharlal Nehru Technological University, Kakinada (JNTUK). He is the Principal Investigator for a Department of Science and Technology (DST)-sponsored project and Co-Principal Investigator (Co-PI) for another DST project. He has published more than 50 research papers in reputed peer-reviewed, Scopus-indexed journals. He has also attended and presented research papers at various national and international conferences, with the proceedings indexed in Institute of Electrical and Electronics Engineers (IEEE) and Springer Link series. He visited Kuching, Malaysia, to attend and present his research articles. Dr. Bujji Babu has published three patent journals, with more pending for grant. He has written more than a dozen monographs, published by technical publishers, and has published two course content modules for the students of Acharya Nagarjuna University. His areas of interest include Software Engineering, Data Mining, Data Science, Big Data, and Programming Languages.