Measurement: Sensors 30 (2023) 100911

Contents lists available at ScienceDirect

Measurement: Sensors

journal homepage: www.sciencedirect.com/journal/measurement-sensors

Improving distributed denial of service attack detection using supervised machine learning

Afrah Fathima [a,b,*], G. Shree Devi [a], Mohd Faizaanuddin [c]

[a] Dept.of Computer Applications, BSAR Crescent Institute of Science and Technology, Chennai, India
[b] Dept. Of CS & IT, Maulana Azad National Urdu University, Hyderabad, India
[c] Dept. of AI & Data Science, M.Tech Chaitanya Bharathi Institute of Technology, Hyderabad, India

ARTICLE INFO

Keywords:
Machine learning
Cyber security
DDoS Attacks
Random Forest Algorithm
Classification Algorithms

ABSTRACT

Distributed denial-of-service (DDoS) attacks are a big problem for cyber security because they can cause a lot of damage to both people and companies. Distributed Denial of Service (DDoS) attacks have been seen to do a lot of damage to the networks and devices they are aimed at. These hacks slow down networks and use up buffer space, which makes resources unavailable. To solve this problem, "Supervised Machine Learning Models" have been used. Several machine learning techniques, such as Random Forest, K-Nearest Neighbors (KNN), and Logistic Regression, were used to figure out what was normal and what was an attack. This study used a sample of the CSE-CICIDS2018, CSE-CICIDS2017, and CICDoS datasets. The dataset was divided into two parts in which three fourth of the data was used for training and one fourth of the data for testing purpose. The proposed research attempt to classify the DDoS attack by using supervised machine learning classifiers. This approach employs three machine learning classifiers such as Random Forest, KNN and Logistic regression. Then we perform Feature Scaling by using Standard Scaler. Finally, the system was evaluated. Random forest classifier outperformed other classifiers with an accuracy of 97.6 % whereas KNN and Logistic regression achieved 97 % and 91.1 %. The study employed several Supervised Machine Learning techniques, including Random Forest, KNN, and Logistic Regression to identify the most effective algorithm for the test. Results demonstrate that Random Forest outperformed the other models.

1. Introduction

Distributed denial-of-service (DDoS) attacks occur when an overwhelming flood of traffic from multiple sources disrupts a victim's service. These attacks pose a significant threat to the Internet. Cybercriminals employ DDoS attacks as powerful weapons, bombarding the victim's network with an immense number of packets. This depletes resources, slows down the service, and prevents legitimate users from accessing it. With the growing number of internet users, these attacks have increased in frequency. They not only block access to services for legitimate users but also result in substantial financial losses. DDoS attacks, which use IP spoofing to complicate request handling and hinder proper operation for genuine users, are the most common form of distributed network attacks. Web applications and commercial websites often become targets of these attacks, which can serve various malicious purposes [1]. DDoS attacks are caused by traditional networking devices' limitations. The ever-growing array of tools and techniques has elevated DDoS attacks to become one of the most significant challenges when it comes to identification and detection. Effectively mitigating DDoS attacks involves the implementation of a comprehensive security policy, deployment of intrusion detection systems, and utilization of firewalls (see Tables 1–4, Figs. 3, 4, 8–14).

DDoS attacks are malevolent attempts aimed at overload a network or website. These assaults present a substantial peril to the stability of the Internet. They involve inundating a network with traffic from numerous origins, thereby exposing users to vulnerabilities. The increased popularity and simplicity of One of the biggest Internet security threats is DDoS attacks. Various machine learning methods have been devised to reduce such threats.

In Fig. 1, the depicted scenario showcases a DDOS attack where an assailant bombards the victim server with botnet attacks, resulting in network disruption. The rapid digitalization of transactions, such as online purchases, testing, e-commerce, online banking, and data transmission, has shifted the majority of activities from the physical realm to

[*] Corresponding author. Dept.of Computer Applications, BSAR Crescent Institute of Science and Technology, Chennai, India.
E-mail addresses: af.fathima1@gmail.com (A. Fathima), Shreedevi@crescent.education (G.S. Devi), mohdfaizaan06583@gmail.com (M. Faizaanuddin).

https://doi.org/10.1016/j.measen.2023.100911
Received 4 August 2023; Received in revised form 17 September 2023; Accepted 7 October 2023
Available online 18 October 2023

Table 1
DDoS attack state-of-art techniques.

S. No	Name of the Technique	Description
1	Botnets	A collection of compromised computers remotely controlled and accessed by an attacker to launch a coordinated attack.
2	Amplification attacks	Using a reflector server to amplify the attack traffic and inundate the target.
3	Application layer attacks	Targeting specific application services, such as HTTP to overload the target.
4	AI- powered attacks	Using machine learning algorithms to make attacks more sophisticated and harder to detect.
5	Encrypted attacks	Using encrypted traffic to hide the attack and evade security measures.

Table 2
DDoS defense approaches.

S. No	Name of the Defense approach	Description
1.	Traffic filtering & rate limiting	limits traffic to identify and stop malicious activity while enabling genuine traffic to reach the target system.
2.	Network infrastructure hardening	This involves implementing security measures in the network infrastructure, such as firewalls, intrusion detection to prevent DDoS attacks.
3.	Traffic classification	This involves classifying network traffic into different categories such as normal, suspicious, and malicious to help identify and mitigate DDoS attacks
4.	Decentralized defense	This involves distributing the defense mechanism across multiple network nodes to prevent any node of failure and improve the overall resilience of the network.
5.	Cloud-based defense	This involves using cloud-based platforms and services to mitigate DDoS attacks.

Table 3
Machine learning defense techniques.

S. No	Machine learning Defense techniques	Description
1	Traffic classification	Machine learning algorithms can be used to categorize network traffic into normal, suspicious, and malicious categories.
2	Anomaly detection	ML algorithms can not only be used to detect anomalies in identifying network traffic patterns but also used in identifying & mitigating DDoS attacks.
3	Attack prediction	ML algorithms can be used to predict potential DDoS attacks based on historical data, allowing for proactive defense measures to be taken.
4	Attack response automation	ML algorithms can be used to mechanize the process of responding to DDoS attacks, reducing the time and resources essential to respond to these threats.

Table 4
Attributes of dataset.

S. No	Name of the feature	Description
1.	Time Stamp	The Date & Time of the Network Traffic event
2.	Fwd pkt Len	The Mean length of the forward packets in the network flow
3.	Fwd Seg Size Avg	The average segment size of the forward packets in the network flow
4.	Init Fwd Win Bytes	The initial size of the forward window in bytes
5.	Init Bwd Win Bytes	The initial size of the backward window in bytes
6.	Fwd Seg Size Mi	The minimum segment size of the forward packets in the network flow
7.	Label	A binary classification of the network traffic as normal or attack

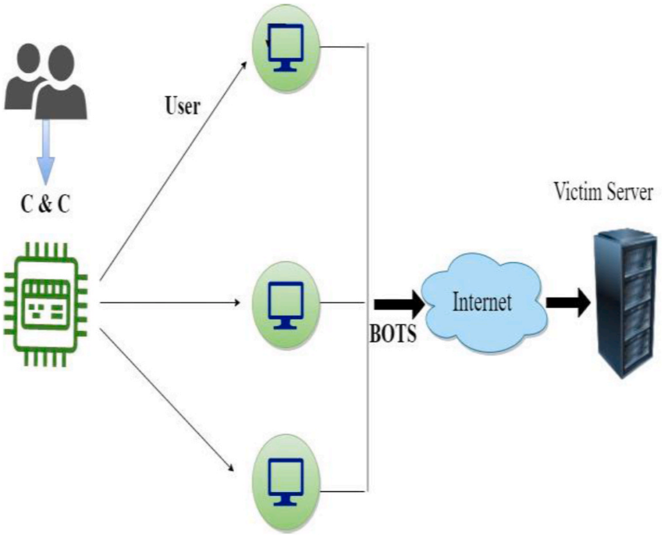


Fig. 1. DDoS attack scenario.

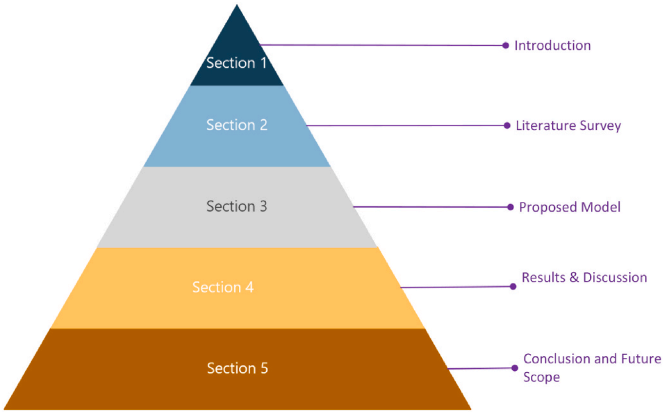


Fig. 2. Organization of the report.

online platforms, thanks to technological advancements [2–5]. The widespread usage of mobile phones enables individuals to participate in this global network, conducting transactions at any time and from any location. However, this digitization also attracts malicious intruders who exploit the Internet's veil of anonymity to pilfer data, money, and disrupt network services. The three primary types of DDoS attacks include application-level attacks, protocol-level attacks, and volumetric attacks [6,7] (see Fig. 2).

Among these algorithms, A-Means and C4.5 hold prominence, each serving distinct purposes [8]. A-Means is an unsupervised clustering algorithm employed to group network traffic data, thereby discerning patterns of normal behaviour and segregating them from anomalous behaviour that may indicate a DDoS attack. Conversely, C4.5 can be utilized to construct a decision tree model that classifies data traffic as either normal or indicative of a DDoS attack [9–11]. While K-Means is commonly utilized for exploratory data analysis, C4.5 is more suitable when the target variable is categorical. C4.5 can be integrated into a comprehensive DDoS attack detection system; however, it may possess limitations and should be combined with other techniques and algorithms for effective detection and in our research, we have employed various supervised machine learning algorithms to identify DDoS

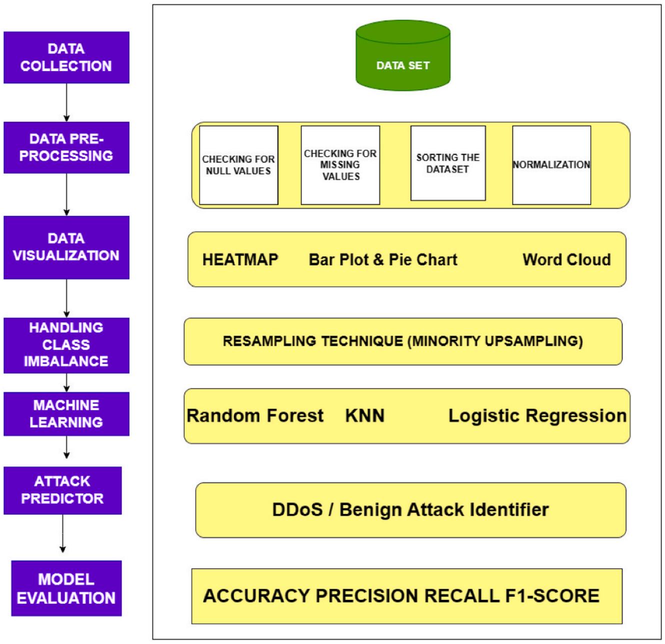


Fig. 3. Architecture of the proposed framework.

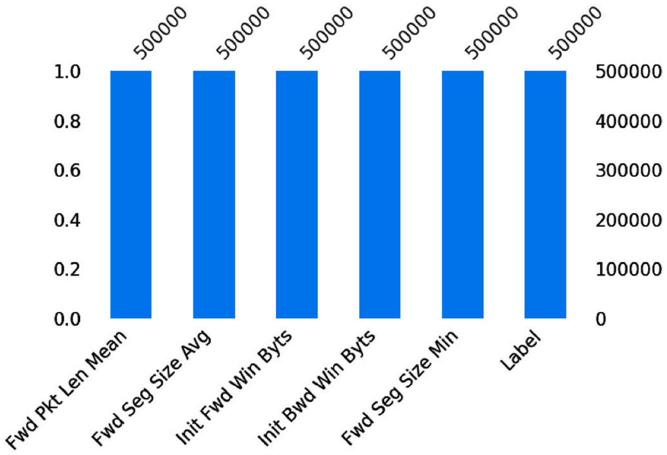


Fig. 4. Checking for Missing values using Missingno.

attacks, evaluating their performance based on accuracy and the Confusion Matrix, considering the severity and increasing frequency of these attacks. To determine the optimal characteristics for DDoS detection, our approach combines supervised learning methodologies with feature selection techniques. Recognizing the serious threat posed by DDoS attacks, we propose a supervised machine learning strategy that integrates different algorithms to enhance the accuracy and efficiency of DDoS detection.

1.1. Motivation

The motivation for research on DDoS attacks using supervised learning is to augment the efficiency of detecting these attacks in real time. Additionally, with the increasing sophistication of DDoS attacks, traditional methods of detecting them may not be effective and consequently, the use of machine learning provides a promising alternative result. This method is tested using real-world data. The below table depicts the various DDoS attack techniques and its defence approaches.

The various DDoS defence approaches are classified in the table below.

1.2. Current state of art in DDoS defense

Present state of the art in DDoS defense involves several key approaches, including Machine learning in DDoS attacks. Machine learning is a rapidly growing area of investigation and development in the ground of DDoS attack defense. Machine learning algorithms can be benefit able in several ways to improve DDoS defense including:

1.3. Objective of research

The primary aim of this study is to investigate the substantial risk presented by Distributed Denial of Service (DDoS) assaults within the domain of cybersecurity, which can have profound consequences for both individuals and entities. DDoS attacks have the capacity to damage target networks and devices through the implementation of overwhelming traffic, resulting in network slowdowns and buffer overflows. Consequently, these assaults render vital resources inaccessible. In order to address this potential risk, the study suggests the adoption of a Supervised Machine Learning framework.

2. Literature survey

Here in this section, we analyze a range of research works done in this field to detect and classify DDoS attacks.

To lessen the effects of DDoS attacks [1], created an algorithmic system built on C.4.5 for detecting them. This approach, when used in conjunction with signature detection methods, produces a decision tree that can be used to automatically and effectively detect signatures assaults used in DDoS floods. The authors have chosen an additional machine learning method and compared the results to verify the accuracy of this system. To detect and track network attacks, the authors trained a recurrent deep neural network on sequences of network traffic. The experimental outcomes show that this model outperforms the modern models for machine learning. In the larger dataset, they have lowered Compared to the conventional machine learning technique, the error rate decreased from 7.517 % to 2.103 % [12].

This study offers ArOMA, an autonomic DDoS defence system that uses SDN's programmability and centralization. ArOMA can automate allowing humans from difficult interventions through traffic monitoring, anomaly detection, and mitigation. By logically spreading security functions, it allows ISPs to manage DDoS traffic in accordance with client demands. The experiments show that ArOMA can maintain video stream performance during DDoS flooding attacks [6]. In this research, the authors have analysed Mirai's seven-month expansion to 600k infections and its DDoS victims. They have also analysed a variety of measuring vantage points were used to examine how the botnet came into existence, what devices were impacted, and how Mirai versions changed and vied for vulnerable hosts. The measurements show how vulnerable the IoT ecosystem is. The authors suggested technological and nontechnical actions and future research to mitigate this danger [13]. DGA-based botnets are hard to detect and survive because to their stealth. This work presents DBod, a DNS query behavior-based DGA--based botnet detection system. The suggested approach leverages the fact all hosts with the same DGA-based malware infection do the same domain list queries, most of which are unsuccessful. The proposed method is tested using 26-month DNS data from an educational network. In actual networks, DBod successfully recognises both recognised and unrecognised DGA-based botnet patterns [14]. This article introduces Zombie Coin, a Bitcoin-based botnet C&C method. Zombie Coin has several advantages over current C&C systems, including its resistance to takedown efforts and regulatory processes. The authors outlined how the Bitcoin network's cutting-edge C&C tools, such as flexible rendezvous scheduling, effective botnet segmentation, and fine-grained bot control, significantly increase this risk. Zombie Coin bots deployed and controlled through the Bitcoin network prove our claims. We predict botmasters will soon seek Bitcoin-based C&C

solutions. The authors aims that this research would help develop effective countermeasures for this menace [15].

This work introduces a unique flow-table sharing strategy to protect SDN-based clouds from flow table overloading DDoS attacks. The switch's flow-table is protected from overloading by using other OpenFlow switches' idle flow-tables. This solution boosts cloud system DDoS resilience with minimal SDN controller participation. Thus, communication overhead is low. Extensive simulations support the authors claims [16]. This research provides a Fuzzy self-organising the ideal solution for enhancing SDN capabilities in cloud computing is maps based DDOS mitigation (FSOMDM). FSOMDM substitutes Kohonen neurons with fuzzy rules. This approach uses software-oriented traffic analysis and the fuzzy rule to explore input space and create a single-valued output for DDoS mitigation. FSOMDM's attack-response process drops attack flows in SDN's control plane. FSOMDM improves true positive rate (TPR) classifier accuracy by about 94 % [17]. This work introduces a structure analysis-based system to categorize botnets and benign applications utilizing traits and patterns linked to the characteristics of botnets. In real-world benchmark datasets, the chosen patterns exhibit strong detection accuracy. The SVM classifier outperforms other classification algorithms in experimental and statistical tests [18]. This work proposes a mathematical a DDoS attack model. Naive Bayes and Logistic Regression algorithms recognise assaults and typical scenarios. Experiments use CAIDA 2007 dataset. This dataset trains and validates machine learning methods. This study implements Weka data mining platform and compares findings. Comparing denial of service attack machine learning techniques with the existing work [19]. Current intrusion detection technologies are unlikely to stop advanced botnet techniques. A botnet traffic analyzer built on deep learning is called Botnet Traffic Shark (BoTShark). BoTShark employs It can handle encrypted payloads since it just handles network transactions and is not dependent on deep packet inspection. This enables us to extract new features from each layer of an autoencoder or cascading convolutional neural networks (CNNs) and uncover connections between the original properties. The authors forecast fraudulent traffic using a Softmax classifier [20].

3. Proposed model

The aim of this research work is to build a framework which can classify and predict DDoS attacks using ML techniques. This framework consists of six steps which include Selecting a suitable dataset, choosing appropriate tools and programming languages, pre-processing the data to eradicate useless information and scale the features, and to visualize the data to gain insights, splitting the data into training and testing sets, building the model, and evaluating it. Machine Learning algorithms named KNN, Random Forest, and Logistic Regrssion have been used for detecting ddos attack and benign scenarios.

3.1. Data set

The dataset used in this study, is a subsample of the CSE–CIC–IDS2018, CICIDS2017, and CICDoS databases (2017). The CSE–CIC–IDS2018, CICID2017, and CICIDS2017 datasets are commonly applied in the thrust of network security to perform intrusion detection and network traffic analysis. Seven columns and 500,000 rows make up this dataset. It has 80 % benign traffic and 20 % DDoS traffic. This dataset contains various features that describe DDoS attacks, including the Timestamp, Forward Pkt Len Mean, Forward Seg Size Avg, Init Fwd Win Bytes, Init backward Win Bytes, Forward Seg Size Min, and the label indicating the type of attack.

3.2. Data pre-processing

In this step, raw network-traffic data is processed and transformed into a pattern that is suitable for analysis by machine learning algorithms. Initially, we checked whether the obtained datasets contain any missing values or not. From the below chart, we can conclude that there are no missing values. X-axis depicts the name of the columns of the data set whereas the y-axis depicts the count of the variables. The sum of rows in the dataset is 5,00000 and seven columns. Statistical techniques is applied to identify and replace irrelevant values in the data. Using the python library "missingno", we visualized the presence and distribution of missing data in the dataframe. The barplot from missingno showed the extent of missing values and their correlation.

3.2.1. Checking for null values

The dataset was checked to know whether the data contain any null values or not. We used isnull() function for this and the results showed that the dataset is free from null values.

3.2.2. Checking for Missing values

Missing values were checked using missingno library and the result can be seen in figure which shows that there are no missing values in the dataset. Using the python library "missingno", we visualized the presence and distribution of missing data in the dataframe. The barplot from missingno showed the extent of missing values and their correlation.

3.2.3. Sorting the dataset

After this we sorted the data using the Timestamp attribute, then the Timestamp column was dropped as it is just a digital record of the time of occurrence of the event and it does not have any impact on the independent variable.

3.2.4. Normalization

In machine learning, feature scaling is used to normalize the distribution of independent variables in a dataset. To prevent some features from overwhelming others due to their scale, feature scaling aims to put all features on an equal scale. As a result, some machine learning algorithms perform better. We have used the Min-Max normalization technique to perform Feature Scaling where the values are transformed so that the mean of the attribute becomes zero and the standard deviation is 1. The result is a normalized distribution which makes it simpler to interpret the relationships among variables. The KNN algorithm, which uses the Euclidean distance as a metric to calculate similarities, can be sensitive to the scale of the features. If some features have a much larger scale than others, the Euclidean distance can be dominated by those larger-scaled features, leading to the suboptimal performance of the algorithm. To address this issue, we have performed feature scaling.

3.3. Data visualization

The representation of data in the form of images or diagrams makes it easier to comprehend and understand the information. To understand and explore the relationships between attributes in the dataset, we employed data visualization using the Seaborn library. We started by creating a heatmap as shown in Figure which presents the data in 2-dimensional color maps utilizing variations in hue, saturation, and luminance. Instead of numbers, heatmaps depict the relationship between variables through colors, with the variables plotted on both axes. The change in color intensity in a particular block represents the relationship between two values.

Fig. 6 represents the heat map which is used for data visualization to show the relationship between two or more variables in a matrix format (see Fig. 5). The color of each cell of the matrix exhibits the level of accordance among the attributes, with warm colors indicating a higher relationship with each other and cool colors indicating a low relationship. The correlation matrix is converted into color labeling via the heat map. In this specific case, the corr_matrix represents the pairwise correlations between numerical columns in the DataFrame 'df'. The code sets the size of the figure for the heatmap and customizes its appearance. The heatmap itself is created using Seaborn's sns.heatmap() function. To

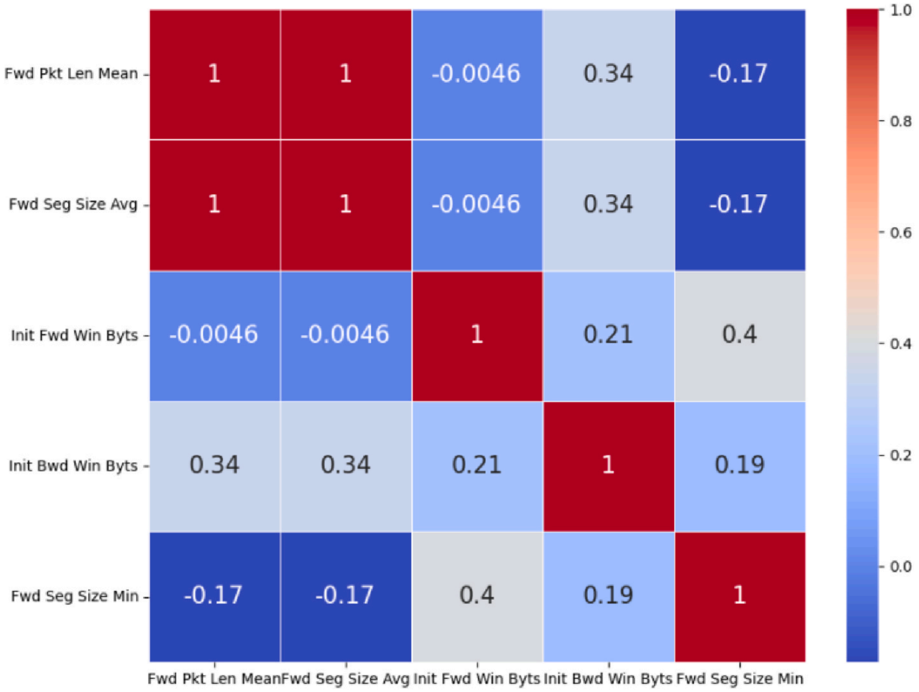


Fig. 5. Heatmap.

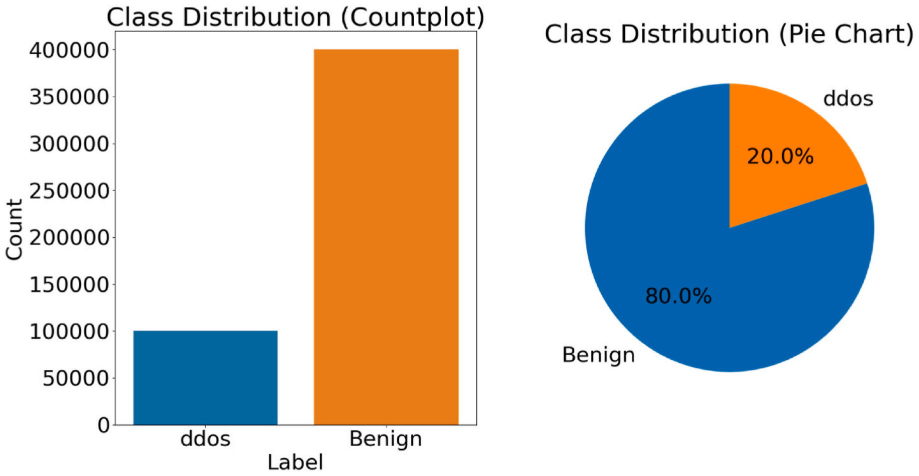


Fig. 6. Count plot and Pie chart of labelled column.

Fig. 7. Word cloud of label column.

enhance visibility, the annot_kws parameter is used to increase the font size of these annotations to 14. The color palette for the heatmap is set to 'coolwarm', which provides a spectrum of colors ranging from cool (blue) to warm (red), making it easier to interpret positive and negative correlations (see Fig. 7).

Count plot shows the distribution of the classes present in the label column. The figure shows that the ddos contains 100,000 values whereas Benign contains 400,000 values. We have used a pie chart which represents the percentage of Bening & DDoS attacks which shows 80 % of benign attacks and 20 % of DDoS attacks.

3.4. Handling class imbalance

Ensuring a balanced distribution of classes within a dataset is crucial, particularly in situations when there is a substantial class imbalance. For instance, in the case where there are only 100,000 DDOS records compared to 400,000 innocuous records, achieving class balance is

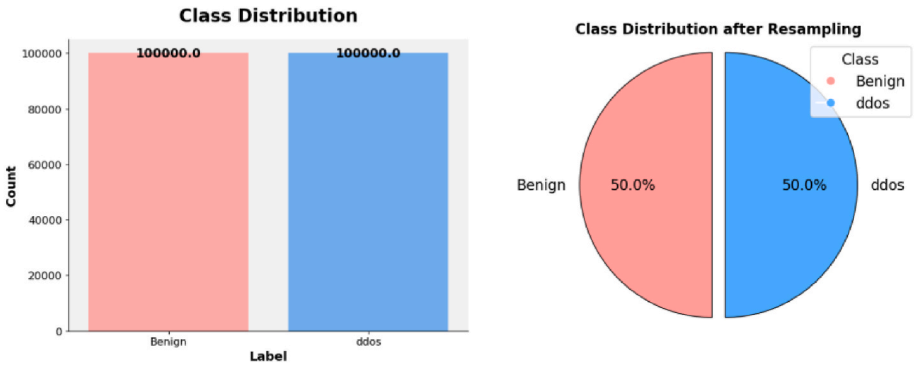


Fig. 8. Class Distribution of Resampling the data.

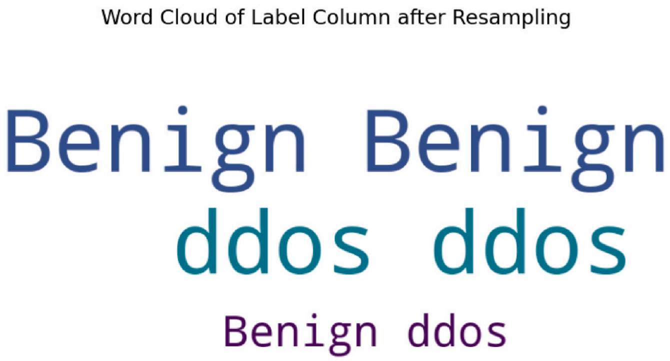


Fig. 9. Word Cloud of Label column after Resampling.

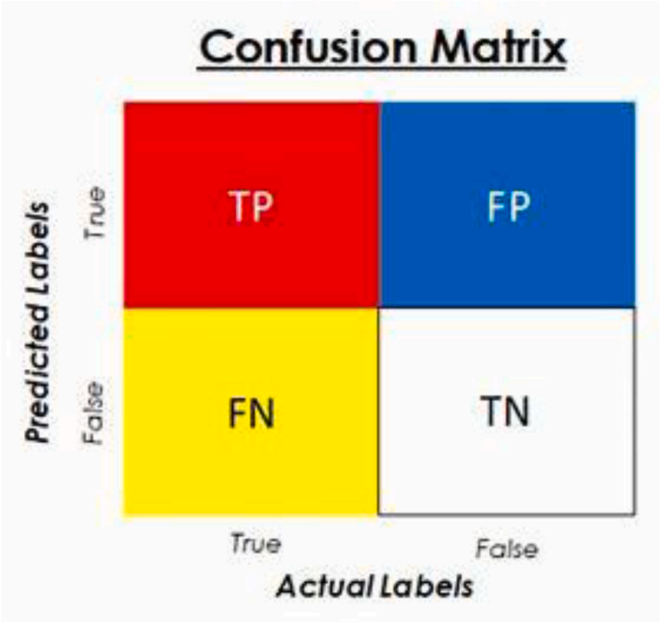


Fig. 10. Confusion matrix.

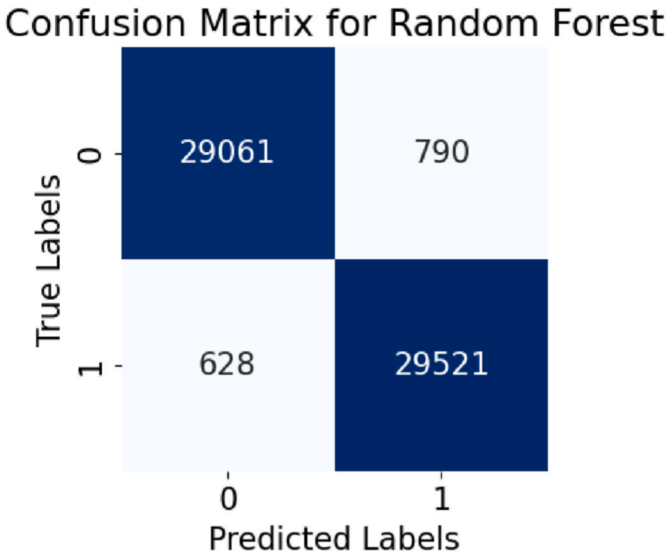


Fig. 11. Confusion matrix of random forest.

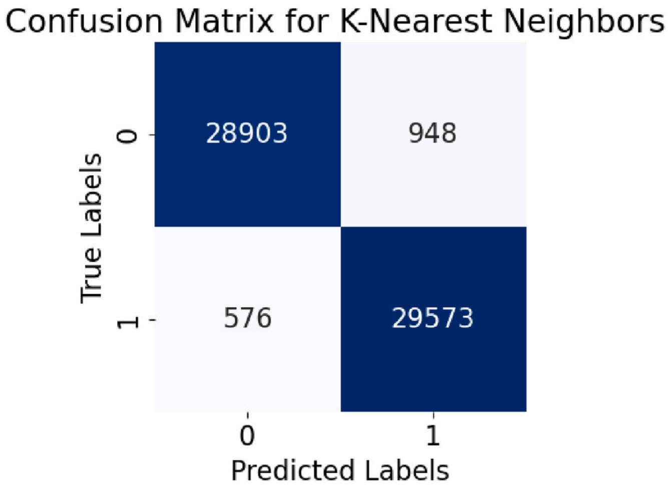


Fig. 12. Confusion matrix of KNN.

necessary to promote equitable and efficient training and performance of machine learning models. The presence of class imbalance might potentially introduce bias, distort outcomes, and adversely impact the generalizability of the model. These concerns can be mitigated by implementing techniques such as under sampling the dominant class. Under sampling is a technique employed to mitigate the issue of models excessively prioritizing the majority class, thereby enhancing the performance pertaining to the minority class. This is particularly relevant in scenarios where the minority class holds significant importance. Moreover, the presence of balanced classes contributes to enhanced model interpretability and equitable evaluation, hence yielding outcomes that are more dependable and significant.

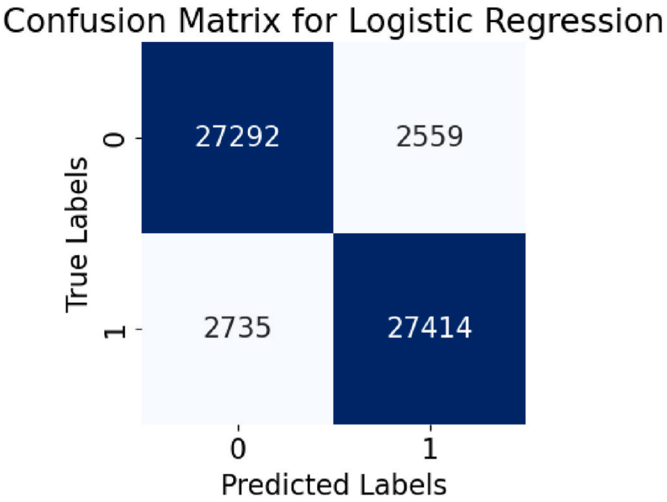


Fig. 13. Confusion matrix of logistic regression.

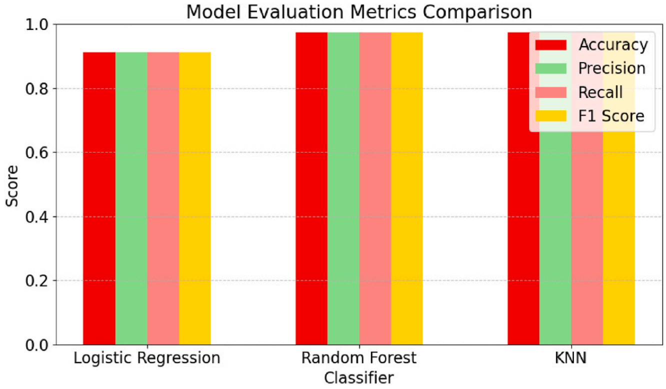


Fig. 14. Accuracy comparison of the Models applied.

3.5. *Data splitting and applying different models*

We have categorized the dataset into two groups: independent and dependent. The target class is another name for the dependent class. Classes that are independent of other classes are referred to as independent classes. For our proposed model, we have partitioned the dataset. To train and test the dataset for evaluation, we have utilized the sklearn model selection library for data splitting. In this research, we have split the data set into two parts the first part which is of 80 % is used for training and the remaining 20 % is used for testing the data. Three different models have been used those are Random Forest, K-Nearest Neighbor, and Logistic Regression.

3.5.1. *Random forest*

Many different trees make up the forest. In the random forest, it has been found that there is no correlation among the decision trees. When a fresh sample input enters the forest after it has been created, the decision trees in the forest will make judgments and decide the sample category based on the decision. A decision tree is used as the classifier in RF, which combines the positive aspects of the random subspace technique and the bagging algorithm. Only a portion of the samples is chosen for inclusion in training sets when the Bagging algorithm is employed for sampling without replacement during training. Decision tree voting is used to determine the outcome.

3.5.2. *K- nearest neighbor*

A Machine Learning method named K-Nearest Neighbor (KNN) is used to classify depending on the mass of their nearest neighbors when the nearest neighbor coverage is set to k.

k value of 5 was selected. This choice was made with several considerations in mind. First, setting k to 5 aims to balance the trade-off between bias and variance. Smaller k values increase sensitivity to noise and may lead to overfitting, while larger k values introduce more smoothing and potential bias. Additionally, an odd k value was chosen to avoid ties in the decision-making process. The selection of k = 5 is a common starting point for many datasets, providing a reasonable balance between capturing local patterns and generalizing well. An odd k parameter in the KNN algorithm has various benefits. First and importantly, it prevents vote ties, ensuring a clear majority among nearest neighbors when classifying. This streamlines decision-making, resulting in clearer results. In imbalanced classes, odd k numbers prevent the majority class from dominating the choice, resulting in a more equal assessment of both classes. Odd k values also smooth feature space decision boundaries, making them ideal for noisy data or outlier reduction.

3.5.3. *Logistic regression*

The objective of this algorithm is to determine the probability of the outcome being either 1 or 0. In contrast to linear regression, which is used to predict continuous outcomes, logistic regression is employed to model the chance of an observation being assigned to one of two distinct groups. The sigmoid function is utilized to convert a linear combination of input data into a probability value ranging from 0 to 1. Logistic regression is subject to certain assumptions, such as the requirement of a linear relationship between the input and output variables, as well as the assumption of independent errors. Regularization methods like L1 and L2 can reduce overfitting.

4. Results and discussion

Machine learning model performance has been assessed using several factors.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{1}$$

Accuracy for Random Forest = 29061 + 29,521 / (29061 + 790 + 628 + 29521) = 0.976

The Accuracy of Random Forest = 97.6%

Accuracy for KNN = 28903 + 29573 / (28903 + 948 + 576 + 29573) = 0.974

Accuracy of KNN = 97.4%

Accuracy for Logistic Regression = 27292 + 27,414 / (27292 + 2559 + 2735 + 27414) = 0.911

The Accuracy of LR = 91.1% (See Table 5)

5. Conclusion and future scope

In this study, we looked at the important problem of Distributed Denial-of-Service (DDoS) attacks, which are a big problem for both people and businesses. We used supervised machine learning to make a System that can tell the difference between normal network traffic and DDoS attacks. We used examples from the CSE-CICIDS2018, CSE-

Table 5
Comparative results of all models.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Logistic Regression	91.1	91	91	91
Random Forest	97.6	98	98	98
KNN	97	97	97	97

CICIDS2017, and CICDoS datasets and split them into training and testing sets. Three machine learning classifiers: Random Forest, K-Nearest Neighbors (KNN), and Logistic Regression have been used. The dataset was carefully checked for null and missing values. This made sure that the study would be based on good data. Scaling the features with Min-Max normalization was an important part of the cleaning step that brought all the features to the same level. This step is especially important for algorithms like KNN that use distance measures and can be sensitive to how big or small the features are. Additionally, we addressed class imbalance issues to ensure that our model learned effectively from both minority and majority classes, enhancing the model's performance and interpretability.

Results demonstrated that Random Forest outperformed other classifiers, achieving an impressive accuracy rate of 97.6 %, followed closely by KNN at 97 %, and Logistic Regression at 91.1 %. These results highlight the effectiveness of the Random Forest algorithm in identifying DDoS attacks in network traffic.

In future we will improve this model by using large and more real time traffic and also aims to improve the accuracy of this models.

Research involving human participants and/or animals

There was no involvement ofhumans and animals in the research.

Informed consent

There was no involvement of human participants in the research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

i have attached already.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.measen.2023.100911.

References

[1] D. Kaur, P. Kaur, Empirical analysis of Web attacks, Proc. Comput. Sci. 78 (2016) 298–306.

[2] M.N. Islam, M. Seera, C.K. Loo, A robust incremental clustering-based facial feature tracking, Appl. Soft Comput. 53 (53) (2017) 34–44.

[3] P.G. Jeya, M. Ravichandran, C. Ravichandran, Efficient classifier for r2l and u2r attacks, Int. J. Comput. Appl. 45 (21) (2012) 28–32.

[4] J. Kang, S. Oh, Anomaly intrusion detection based on clustering a data stream, Int. J. Future Comput. Commun. 1 (1) (2012) 17–20.

[5] G. Kim, S. Lee, S. Kim, A novel hybrid intrusion detection method integrating anomaly detection with misuse detection, Expert Syst. Appl. 41 (4) (2017) 1690–1700.

[6] K. Ramasubramanian, A. Singh, Machine Learning Theory and Practices," *Machine Learning Using R*, Apress, Berkeley, CA, 2017. Detecting Denial of Service attacks using machine learning algorithms Kimmi Kumari* and M. Mrunalini.

[7] W. Feng, Q. Zhang, G. Hu, J.X. Huang, Mining network data for intrusion detection through combining Svms with ant colony networks, Future Generat. Comput. Syst. 37 (2014) 127–140.

[8] A.P. Muniyandi, R. Rajeswari, R. Rajaram, Network anomaly detection by cascading k-means clustering and C4.5 decision tree algorithm, Procedia Eng. 30 (2012) 174–182.

[9] R. Cheng, R. Xu, X. Tang, V.S. Sheng, C. Cai, An abnormal network flow feature sequence prediction approach for DDoS attacks detection in the big data environment, Comput. Mater. Continua (CMC) 55 (1) (2018) 95–119.

[10] O. Depren, M. Topallar, E. Anarim, M.K. Ciliz, An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks, Expert Syst. Appl. 29 (4) (2005) 713–722.

[11] J. Qiu, L. Du, D. Zhang, S. Su, Z. Tian, Nei-TTE: intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city, IEEE Trans. Ind. Inf. 16 (4) (2020) 2659–2666.

[12] S.K. Ajagekar, V. Jadhav, Study on web DDOS attacks detection using multinomial classifier, in: Proceedings of the IEEE International Conference on Computational Intelligence & Computing Research IEEE, December 2016. Chennai, India.

[13] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, K. Lee, Proceedings of the 2008 ACM Conference on Emerging Network Experiment and Technology, Context 2008, December 2008.

[14] R.A. Calix, S. Rajesh, Feature ranking and support vector machines classification analysis of the NSL-KDD intrusion detection corpus, in: Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, May 2013. Palo Alto, California.

[15] M. Panda, A. Abraham, M.R. Patra, A hybrid intelligent approach for network intrusion detection, Procedia Eng. 30 (4) (2012) 1–9.

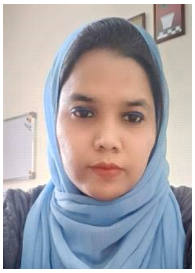
[16] Y. Chen, A. Abraham, B. Yang, Hybrid flexible neural tree-based intrusion detection systems, Int. J. Intell. Syst. 22 (4) (2007) 337–352.

[17] C. Luo, Z. Tan, G. Min, J. Gan, W. Shi, Z. Tian, A novel web attack detection system for internet of things via ensemble classification, IEEE Trans. Ind. Inf. 17 (8) (2021) 5810–5818.

[18] Z. Tian, X. Gao, S. Su, J. Qiu, Vcash: a novel reputation framework for identifying denial of traffic service in internet of connected vehicles, IEEE Internet Things J. 7 (5) (2020) 3901–3909.

[19] J.E. Gaffney, J.W. Ulvila, Evaluation of intrusion detectors: a decision theory approach, in: Proceedings of the 2001 IEEE Symposium on Security and Privacy, May 2001, pp. 50–61. Oakland, CA, USA.

[20] X. Huang, Y. Ye, L. Xiong, S. Wang, X. Yang, Clustering time-stamped data using multiple nonnegative matrices factorization, Knowl. Base Syst. 114 (2016) 88–98.



Ms. Afrah Fathima earned her Bachelor of Computer Applications Degree from Osmania University. Her Masters Degree in Computer Applications from Osmania University and she is pursuing her Ph.D.(Part Time) degree in Computer Applications from the B.S Abdur Rahman Crescent Institute of Science & Technology. She is currently an Assistant professor with the Department of Computer Science & Information Technology in Maulana Azad National Urdu University, Telangana, Hyderabad, India. Her research interests include, Cyber Security, Wireless Networks, Machine Learning, Digital Forensics

Dr. G. Shree Devi completed her Bachelor's Degree in B.Sc Mathematics from University of Madras, in 1999; her MCA Degree in Computer Applications from Periyar University, in 2002. and her Ph.D. degree in Computer Science from the B.S Abdur Rahman Crescent Institute of Science & Technology, in 2018. She is currently an Assistant professor (Sel.Grade)with the Department of Computer Applications, B.S Abdur Rahman Crescent Institute of Science & Technology, Chennai, India. She is the author and coauthor of more than 15 publications. Her research interests include Image Processing, Data Mining, Data Science, Pattern Recognition, Statistical Pattern Classification.

Mohd Faizaanuddin completed his Bachelor of Technology from Keshav Memorial Institute of Technology in Computer Science and is currently pursuing M.Tech from Chaitanya Bharathi Institute of Technology in Artificial Intelligence and Data Science.