IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 4, Issue 2, November 2024

Timely Detection of DDoS Attacks with Dimenstionality Reduction

Miss. Aishwarya Anil Shelke, Miss. Pratiksha Valmik Sonawane, Miss. Kajal Bhausaheb Pathare, Miss. Ishika Vikas Bagore

Department of Information Technology

SND College of Engineering & Research Center, Yeola, Maharashtra, India

Abstract*: Due to the interconnectedness and exponential proliferation of IoT devices, the technology is more susceptible to network attacks like Distributed Denial of Service (DDoS), which disrupt network resources. A growing threat to cloud computing systems is the Distributed Denial of Service (DDoS) attack, in which the attacker starts the attack by taking advantage of computers both inside and outside the cloud system. Real-time analysis of cloud network data is essential for preventing DDoS attacks. DDoS attacks interfere with the operation of Io- connected apps and services by taking advantage of the constrained resources on IoT devices. The impacts of DDoS attacks, which seriously damage current systems, are thoroughly examined in this article in the context of the Internet of Things. One of the most common network attacks is the distributed denial-of-service attack (DDoS). DDoS assaults intensified due to the quick development of computer and communication technologies. Therefore, investigating the detection of a DDoS attack is crucial. A single technique cannot offer adequate security due to the variety of DDoS attack techniques.*

Keywords: Botnet, Cloud Computing, Deep Learning, Distributed Denial-of-Service Attacks,IoT

I. INTRODUCTION

The rapid development of computer and communication technology makes DDoS attacks increasingly severe. Therefore, investigating how to detect a DDoS attack is essential. Because DDoS attacks can take many different forms, no single technique can provide sufficient security. Arthur Samuel created machine learning (ML) methods in 1959, asserting that a machine could learn from historical data in a certain context without explicit programming. Virtual services, in which companies employ cloud computing platforms to offer their products and services digitally, are another facet of artificial intelligence. This provides a comprehensive business infrastructure for the global delivery of internet-based services. The increasing complexity of computer networks is making it difficult for traditional network architectures to meet the demands of contemporary cloud computing, mobile Internet, and other components for scalable and diverse network services. This is due to their fixed form and the strong connection between the control and data-forwarding functions. By separating the network control function from the data forwarding function, SDN, a unique form of network design, provides greater flexibility and programmability than conventional networks. Even while SDN design offers centralized network control and on-demand traffic routing, it still has significant security weaknesses and is more susceptible to security assaults. Denial-of-service attacks, which disrupt SDN's availability, are one common attack method among them. An attacker can use a denial-of-service attack, which involves sending malicious traffic to computer network hosts, to exhaust the network's finite resources and disrupt its availability, preventing it from providing regular services. Attacks known as distributed denial of service happen. DoS attacks try to exploit network protocol weaknesses and the limited availability of network resources by sending out a large number of incorrect data packets. By depleting the network's bandwidth, connection, and service resources, this ultimately prevents authorized users from accessing it. These days, denial-of-service (DoS) assaults are a crucial strategy in cyberwarfare. Russia launched DDoS attacks against numerous military and governmental targets during the conflict between Russia and Ukraine. These days, DoS attacks are a crucial tactic in cyberwarfare. Russia launched DDoS attacks against numerous Ukrainian financial, governmental, and military websites during the conflict between Russia and Ukraine. These attacks caused several critical network systems and infrastructures to fail, which had a significant impact on Ukraine's. In an attempt to protect networks, it includes many techniques, such as statistical-based ones, that collect and analyze flow-related data. They can be applied to a number of situations, such as entropy-based methods, anomaly detection, time-series analysis, and trust management. However, these methods struggle to accurately distinguish between malicious and legitimate traffic and to adjust to emerging threats. This could lead to either false positives or false negatives. Machine learning (ML) algorithms showed a strong potential for detecting and mitigating DDoS attacks. Real-time network traffic analysis, malicious activity detection, and detection model modification in response to real- time network data are all capabilities of these techniques. They can also handle enormous volumes of network data and spot patterns in zero-day assaults. However, depending only on statistical techniques, such as entropy-based detection, may not be accurate or responsive enough, especially in large- scale and dynamic SDN systems. In addition to improving the statistical method, machine learning clustering techniques like the K-means algorithm make it easier to analyze complex patterns and identify odd clusters of network activity. Combining machine learning clustering techniques with system entropy aims to provide a comprehensive description and tackle several significant problems in DDoS attack detection and mitigation, including DDoS attack detection and mitigation in SDN networks.

II. LITERATURE REVIEW

The recent surge in DDoS attacks has drawn a lot of interest from researchers, who are looking for various ways to identify, lessen, and stop them. This section enumerates some earlier research that sought to address these attacks by proposing various strategies.

In order to mitigate DDoS attacks in cloud systems, Mohammed H. Sqalli et al. [6] proposed the EDoS-Shield solution, which relies on black lists created by suspicious IP addresses identified by earlier verification procedures. The system firewall then blocks any incoming IP address that is included in these lists.

Using bloom filters and drawings, Chenxu Wang et al. [7] created the SkyShield technique to identify DDoS attacks on a web server's application layer. The mitigation phase's bloom filters classify incoming requests into black and white lists based on CAPTCHA methods. During the detecting phase, sketches define the malicious IP address's anomalous sketch based on the amount of requests and its overload. Using flow data and time series, Anna Sperotto et al. [8] developed their detection method, concentrating on how the attack affects the time series amount of flows, packets, and bytes.

These conventional methods rely on particular network conditions and may be impacted by the actions of the attackers. Furthermore, the black and white lists solutions are no longer adequate due to new attack strategies such using fake IP addresses and reflecting the attack from many valid IP addresses. Additionally, DDoS attacks target both the network and application layers.

In addition to just identifying specified attacks and requiring constant human intervention, previous intrusion detection systems (IDS) lacked flexibility in detecting DDoS attacks. As a result, researchers have to stay up to date with developments and seek out new approaches. Some researchers have concentrated on machine learning (ML) techniques, employing their classifiers in a variety of solutions; these techniques rely on an effective dataset of network transactions to train the ML classifier to detect the assault. It includes both DDoS and innocuous traffic, each of which can be identified by a variety of characteristics. The gathered dataset, the characteristics that are extracted, and the classifier that is used all affect how effective machine learning techniques are.

To identify the IoT device causing the DDoS attack, Rohan Dochi et al. [9] introduced the Pipeline framework, a machine learning DDoS detection tool for IoT network traffic. They claimed that there aren't enough datasets in this field, so they generated theirs by mimicking a consumer IoT system network. Four stages make up the Pipeline framework: anomaly detection, traffic capture, packet grouping, feature extraction, and binary classification. In order to utilize the stateless feature's lightweight and real-time intrusion detection and the statefull feature's extremely accurate detection rate, the researchers examined both statefull and stateless capabilities in order to extract the features.

The K-nearest neighbours (K-NN), Decision Tree, Random Forest, and Neural Network classifiers outperformed the Linear Support Vector Machine (LSVM) by 99.1%, 99.9%, and 99.9%, respectively, in their framework's evaluation accuracy. In addition to lowering the computational and financial overheads, the selected classifier algorithms have proven successful in accurately characterizing DDOS traffic for the Internet of Things network. However, in order to

DOI: 10.48175/IJARSCT-22158

minimize overheads that can affect real-time identification, the researchers limit the collection of features they have chosen.

Additionally, they estimated that the DDoS attack lasted less than one and a half minutes, but it might have lasted longer. It would be challenging to execute their framework correctly if such were the case. Using Naïve Bayesian, Decision Tree (C4.5), and K-Means classifiers in a virtual machine and virtual LAN, Marwane Zekri1 et al. [1] focused on DDoS attacks that depleted the capacity of cloud systems. Their findings showed that the decision tree (C4.5) achieved 98.8%, K- Means obtained 95.9%, and the Naive Bayesian performance rate equaled 91.4%. In summary, of the classifiers presented, the decision tree (C4.5) with the highest accuracy also has the shortest running time.

In order to take advantage of the excellent accuracy and low false positive rate of supervised and unsupervised techniques, researchers on Mohamed Idhammad et al. selected anomaly detection using semi- supervised Ml. Prior to the supervised algorithms properly classifying this data to lower the false positive ratio, the unsupervised algorithms decreased The accuracy of the co-clustering classifier was 98.23% while using the NSL-KDD dataset, 99.88% when using UNB ISCX IDS 2012, and 93.74% when using UNSW-NB15. Additionally, the Extra-Trea Ensamble classifier's accuracy was 82.73% while using the NSL-KDD dataset, 61.22% when using UNB ISCX IDS 2012, and 86.56% when using UNSW-NB15. The researchers asserted that the UNSW-NB15 data set's low ratio was attained due to the significant degree of similarity between intrusion traffic and regular traffic.

Using packet parameters including source and destination addresses, packet counts, and threshold values, K. Gurulakshmi et al. suggested a method to identify malicious packets received from Internet of Things devices. SVM and K-NN were their classifiers for identifying the unusual traffic. In this study, the K-NN had a superior accuracy ratio when the featured set was smaller, equal to 98%, whereas the SVM scored higher when the featured set was larger, equal to 95%. In order to minimize the overhead computation and attain the necessary accuracy, the packet information set was utilized in conjunction with both classifiers for integration.

Jiangpan Hou et al. trained Randon Forest ML classifiers in their suggested approach using the Cisco Netflow dataset to identify the DDoS assault. They employed two sets of features, flow-based and patterns-based, to ensure that the DDoS attack was effectively detected by sorting the most effective features for data sampling. They tested their plan using a public dataset (CIC-IDS2017), a sizable ISP Netflow dataset, and a lab network created by well- known DDoS tools. Over 99% of the results of their experiment were accurate.

III. METHODOLOGY

A. RANDOM FOREST

A group of decision trees make comprisethe ensemble approach known as RF. Classifiers that use the divide and conquer overall bias of a single tree, RF makes use of bagging and randomization. strategy are called decision trees. Everynode separates the data, and the leaves finish the categorization. To aggregate the decision tree forecasts and lessen the K-neighbors in the area Regression and classification problems are addressed by the non-parametric supervised learningalgorithm KNN. A datapoint and the k closest data entry, referred to as neighbours, are chosen by the algorithm. The majority class of the neighbours is used to categorize the new data. In general,an under-t model will result from tiny k values. In addition to increasing bias and computing time, higher values will decrease variance. Selecting the ideal k value is crucial for the classifier to operate at its best.

B. K-NEAREST NEIGHBOUR

Side Sampling (GOSS) and ExclusiveFeature Bundling (EFB) to try to solve thisproblem. EFB creates a single feature out of characteristics that are mutually exclusive. Because larger gradients are typically linked to greater information gain, GOSS randomly removes smaller gradients. The temporal complexity is decreased by combining these two techniques.

C. LIGHTGBM

LIGHTGBM One kind of gradient boosteddecision tree is LGBM, which was developed by Microsoft. Decision trees typically find splits that have the most information gain or the largest change in entropy both before and after each split.

ISSN
2581-9429
IJARSCT

DOI: 10.48175/IJARSCT-22158

The optimal split is determined using apre-sorted or histogram-based approach, which can be quite time-consuming for large datasets.

D. LINEAR DISCRIMINANTANALYSIS (LD)

The linear classification method known as LINEAR DISCRIMINANT ANALYSIS (LD) is generally regarded as superior to logistic regression for multiclass classification. In order to represent featuresfrom a higher dimension in a lower dimension, the algorithm employs a dimensionality reduction technique. Bydoing this, it seeks to identify group distinctions. Separate classes are represented by the distinct groups in lower dimensions.

E. NAIVE GAUSSIAN BAYES

A probabilistic algorithm that adheres to the Gaussian normal distribution, GNB can handle continuous data. GNB is a Naive Bayes extension. The Bayes theorem serves as the foundation for the classification algorithm known as Naive Bayes.

F. SUPPORT VECTOR MACHINE(SVM)

A classifier called a support vector machine looks for a hyperplane in an N- assumption that characteristics are distributed normally and are independent of one another. Results are computed for probability likelihoods using a distribution's probability density function. GNB has the benefit of performing well with little data sets. dimensional space that will divide data points into distinct groups. The SVM selects the plane with the largest marginout of the several possible hyperplanes. The largest gap between the data points of the two classes is represented by the maximum margin. SVMs operate quickly and effectively with little data. In an N- dimensional space, the SVM classifier looks for a hyperplane that will divide datapoints into distinct classes.

The SVM selects the plane with the largestmargin out of the several possiblehyperplanes. The largest gap between the data points of the two classes isrepresented by the maximum margin. SVMs operate quickly and effectively withlittle data. SVM is frequently employed in many different fields. Although it is a strong algorithm, huge datasets may make it computationally costly.

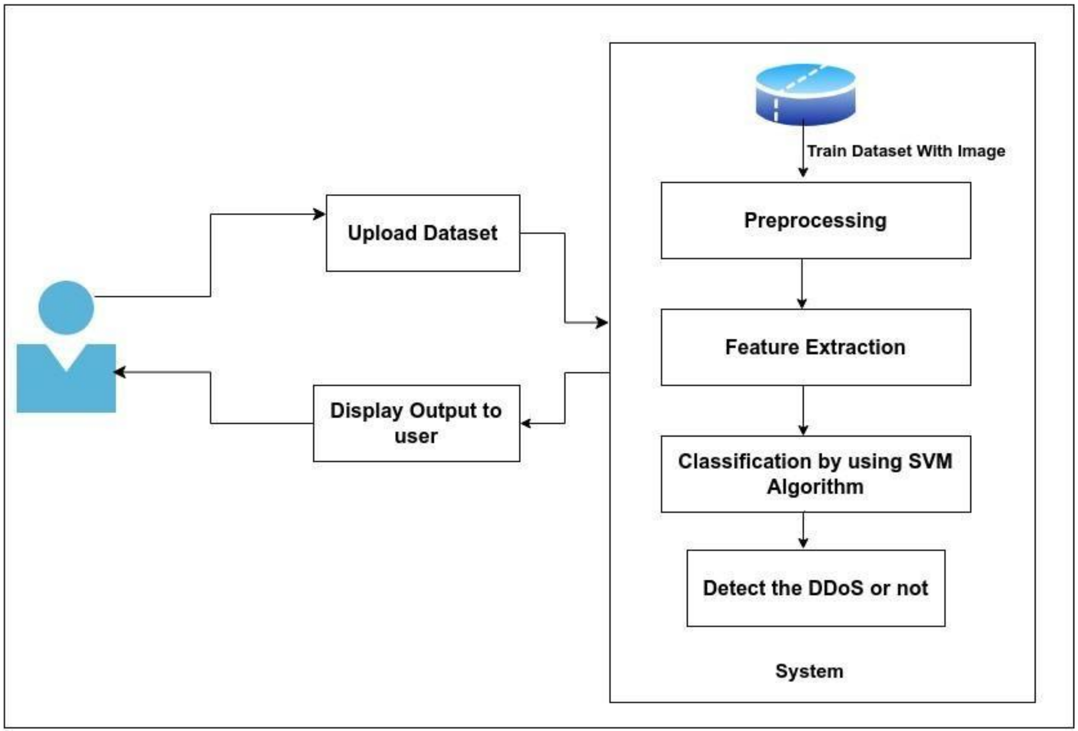

Fig. 1 Structural Diagram

DOI: 10.48175/IJARSCT-22158

SVM can handle high-dimensional feature spaces and is especially good at handling complex datasets. Finding the best hyperplane to divide the data points of several classes in the feature space is the fundamental concept of support vector machines (SVM). The underlying premise of SVM is that the data may be linearly separated. The data points that are closest to the decision boundary (hyper-plane) are known as support vectors. These points are utilized to make predictions and are essentially in defining the hyper-plane. The data points that are closest to the decision boundary (hyper-plane) are known as support vectors.

1) Data pre-processing:

The data was cleaned, trimmed, encoded, and standardized prior to classifier training. Of the 88 features, 18 have to be eliminated for this. Four of the identifying features that would overfit to the classifiers—source and destination ports and source and destination IP addresses— as well as nine features with only zero values were eliminated from the training set. Duplicate or unlabeled features were the other features eliminated. Samples with infinite or missing values were then removed. After the data was divided into training and testing subsets, normalization was carried out to prevent inadvertent bias from being introduced into the dataset.

Down sampling was done to keep training times manageable and get around the memory constraint on a standard desktop computer because the dataset contains more than 70 million samples. In order to balance the extremely unbalanced dataset, which included fewer than 1% benign samples, down sampling was also carried out. On the unbalanced dataset, a classifier might reach over 99% accuracy even if it incorrectly classified every sample as DDoS. The 200K samples that made up the binary classification data were split 50/50 between harmful and benign samples in order to balance the dataset. All 12 DDoS attack types were equally represented in the malicious samples, which were thought to have been chosen at random. Because we are working with real-world network traffic data that is not properly balanced, scaled, and distributed, we must handle this data with the required preprocessing to improve the performance of DEQSVC and produce accurate results. The preprocessing step consists of three parts: data cleaning, data balancing (scaling and normalization), and dimensionality reduction.

TIME-BASED FEATURES

Our time-based dataset consists of 25 features. These features were aggregated by Lashkari et al. for their research, and have proven to be effective in detecting VPN and Tor traffic that DDoS attacks involve sending large numbers of packets over a short period of time, we hypothesize that these time-based features would effectively translate to DDoS detection problems. Each of the first five features marked with a described in the table is further broken down into four separate statistical features: mean, min, max, and standard deviation.

1) DATA CLEANING:

Because the dataset comprises actual data points that depict network traffic, it contains erroneous information that lowers detection accuracy. We cleaned up the dataset in the following ways to address this issue:

1) Remove any data points with white spaces or null values.

2) Using the CICFlowMeter-V3, exclude data points in the initial network traffic analysis that have a missing percentage greater than a predetermined threshold.

3) Eliminate characteristics that don't fit into a formal range distribution and have a single distinct value.

4) Remove data points that are collinear and have a correlation greater than a predetermined correlation coefficient.

2) DATA BALANCING:

The dataset includes unbalanced, informal distribution range data from real-world network traffic. However, the dataset needs to be normalized and standardized in order to improve the DEQSVC model's detection accuracy. Therefore, we balanced our dataset and normalized it to a formal distribution range using the Standard Scaler and Min-Max Scaler procedures. By removing the mean and scaling to a variance unit, the Standard Scaler method standardizes characteristics. The following is the formula for normalizing a dataset sample x. z = (x − v)/s, where is the training dataset's standard deviation and visits mean. Nevertheless, we scale each feature separately by computing the pertinent statistics on the training set's data points. The mean and standard deviation are then saved for use with subsequent data using the transform approach. Once the dataset has been standardized, we use the Min-Max Scaler to minimize its dimensions and convert each one into a range of (−1,1). Since the majority of the characteristics in the chosen dataset had variance in the same order and a center around 0, we standardized and normalized the dataset for the DEQSVC model.

A feature will control the kernel function and prevent the training model from accurately learning from other features, though, if its variance is orders of magnitude greater than that of other features. Thus, by performing the standardization and normalization processes, we increase the detection accuracy of our project model.

3) DIMENSIONALITY REDUCTION:

As previously mentioned, our goal is to provide a QSVC technique with the best cybersecurity DDoS attack detection accuracy. Nevertheless, the dataset's high complexity could result in subpar training times and detection accuracy. In order to get around this problem, we reduced the training dataset's dimensions to two using the quick ICA Dimensionality reduction technique. An improved form of the ICA, the fast ICA is a popular dimensionality technique for breaking down variables with many outcomes into independent sub- components. The sample dataset is represented as a random vector x = (x1, x2, ..., xm) T in the fast ICA approach, and the random states corresponding to the observed variables are represented as a random vector s = (s1,s2,...., sn) T. objective is to convert x into a vector of maximum sub-independent components, estimated by some function F (s1,s2,. ,si)

T of independence, using a linear static transformation. The data points in our dataset are linearly independent. Thus, in order to produce the reduced features, we employed the fast ICA equation as follows: The formula is xi = di,1s1 + + di,k sk +... + di,n$. (1) In this case, di,k represents the data point di's mixing weight, and x = (x1, x2,...., xm). For the random states sk, k = 1, 2, 3,. , n, T is the sum of their independent subcomponents.

Advantages :-

a) Real-time Detection: The model can identify DDoS attacks in real-time, allowing for quick response and mitigation measures, by continuously monitoring system entropy and utilizing machine learning clustering methods.

b) Adaptability:

DDoS attack techniques are always changing; therefore, detection systems must adjust and detect new attack patterns. By using machine learning clustering techniques, the model's resilience against new threats is increased since it may dynamically modify its detection skills in response to observed network behaviour.

c) Scalability:

Several network devices and a variety of traffic patterns are common in SDN systems. The suggested approach is made to be scalable to manage the complexity and volume of network traffic, providing accurate detection and extensive coverage.

IV. FUTURE SCOPE

Even though the DoS2019 dataset includes a wide range of DDoS attack types, many more attacks have not yet been evaluated. Current classifiers might not be able to identify some unproven DDoS attack types, like low-rate DDoS attacks. The flow intervals utilized to create the traffic flows were not disclosed by the DDoS2019 [6] creators. There might be more ideal flow intervals out there that would be better able to identify these attacks. By recreating the tabular dataset from the raw traffic data over various flow intervals using CIC-Flow Meter, more research may be done to determine the ideal flow intervals to enhance the performance of the time-based features. Only KNN used hyperparameter optimization in our experiments. By performing grid search over all classifiers and fine-tuning hyperparameters, future research could potentially improve the models seen in this study. There are other types of deep learning classifiers that merit testing in this field and with the time-based feature set, however for this investigation, a single DNN model was trained and evaluated using a conventional configuration. In order to identify associations between features, several of these classifiers, like CNNs, considerably benefit from the conversion of tabular data to visual representations [16]. Future research could use these more sophisticated classifiers in conjunction with data augmentation approaches to examine time-based attributes.

Our tests' findings show that smaller feature groups can effectively increase model training speed with Experiment with different feature groups and feature engineering could produce a more reliable dataset, which could lead to more efficient models without sacrificing performance. Last but not least, the suggested time-based feature subset appears to be a viable way to reduce training timeframes while maintaining a similar level of accuracy, possibly in applications other than DDoS and Tor classification. since the models are constantly being trained—possibly in real time. The efficiency of the time-based feature set in identifying and categorizing Tor traffic was demonstrated by Lashkari et al. Similarly, our research showed that time-based characteristics are viable in the DDoS arena. The findings in this publication and the work of Lashkari et al. suggest that time-based features should be further tested in traffic-based studies where a smaller dataset can be useful to avoid the problem of overfitting.

V. CONCLUSION

The constant threat of DDoS attacks, which cause enormous amounts of damage and depreciation, has increased the demand for some strong and dependable countermeasures. It is crucial to establish a cohesive strategy for dealing with such an infamous foe. In light of this objective, this study attempts to offer useful insights regarding DDoS attacks and their effects in a variety of Internet domains, from conventional networks to decentralized environments based on blockchain. In this study, we have discussed a wide range of detection strategies that have been developed to date by academics across different disciplines. Although these defense strategies produce commendable results, much more has to be investigated to address the unresolved issues these sectors face. In order to create sophisticated DDoS protection systems, we think that the work described here offers a fundamental knowledge of the problem.

REFERENCES

[1] T. Mahjabin, Y. Xiao, G. Sun, and W. Jiang, ‘‘A survey of distributed denial-of-service attack, prevention, and mitigation techniques,’’ Int. J. Distrib. Sensor Netw., vol. 13, no. 12, Dec. 2017, Art. no. 155014771774146, doi: 10.1177/1550147717741463.

[2] S. S. Mohammed, R. Hussain, O. Senko, B. Bimaganbetov, J. Lee, F. Hussain, C. A. Kerrache, E. Barka, and M. Z. A. Bhuiyan, ‘‘A new machine learning-based collaborative DDoS mitigation mechanism in software-defined network,’’ in Proc. 14th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob), Oct. 2018, pp. 1–8, doi: 10.1109/WIMOB.2018.8589104.

[3] A. Callado, C. Kamienski, G. Szabo, B. P. Gero, J. Kelner, S. Fernandes, and D. Sadok, ‘‘A survey on internet traffic identification,’’ IEEE Commun. Surveys Tuts., vol. 11, no. 3, pp. 37–52, 3rd Quart., 2009, doi: 10.1109/SURV.2009.090304.

[4] X. Ying, ‘‘An overview of overfitting and its solutions,’’ J. Phys., Conf. Ser., vol. 1168, Feb. 2019, Art. no. 022022, doi:10.1088/1742-6596/1168/2/022022.

[5] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, ‘‘Characterization of tor traffic using time based features,’’ in Proc. 3rd Int. Conf. Inf. Syst. Secur. Privacy, 2017, pp. 253–262, doi:10.5220/0006105602530262.

[6] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, ‘‘Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy,’’ in Proc. IEEE 53rd Int. Carnahan Conf.Secur. Technol. (ICCST), Oct. 2019, pp. 1–8. [Online]. Available:https://ieeexplore.ieee.org/abstract/document/8888419

[7] A. Lashkari, ‘‘CICFlowmeter-V4.0 (formerly known as ISCXFlowMeter) is a network traffic Biflow generator and analyser for anomaly detection,’’ Canadian Institute of Cyber Security (CIC), Fredericton, New Brunswick, Tech. Rep., 2019. [Online]. Available: https:// github.com/ISCX/CICFlowMeter, doi.

[8] J. Chen, Y. Yang, K. Hu, H. Zheng, and Z. Wang, DAD-MCNN: DDoS attack detection via multi-channel CNN, Proc. 11th Int. Conf. Mach. Learn. Comput. (ICMLC) 2019, pp. 484488,doi:10.1145/3318299.3318329.

[9] S. S. Priya, M. Sivaram, D. Yuvaraj, and A. Jayanthiladevi, Machine learning based DDOS detection, in Proc. Int. Conf. Emerg. Smart Comput. Informat. (ESCI), Mar. 2020, pp. 234237, doi: 10.1109/ESCI48226.2020.9167642.

[10] O. Elejla, B. Belaton, M. Anbar, B. Alabsi, and A. Al-Ani, Comparison of classi cation algorithms on icmpv6-based DDoS attacks detection, in Computational Science and Technology (Lecture Notes in Electrical Engineering). Singapore: Springer, 2018, doi: 10.1007/978-981-13-2622 6_34.

[11] O. E. Elejla, M. Anbar, B. Belaton, and S. Hamouda, Labeled ow-based dataset of ICMPv6-based DDoS attacks, Neural Comput. Appl., vol. 31, no. 8, pp. 36293646, Aug. 2019. [Online]. Available: https://link.springer.com/article/10.1007/s00521-017- 3319-7

[12] R. F. Fouladi, O. Ermi , and E. Anarim, A novel approach for distributed denial of service defense using continuous wavelet transform and convo lutional neural network for software-de ned network, Comput. Secur., vol. 112, Jan. 2022, Art. no. 102524, doi:10.1016/j.cose.2021.102524.

[13] Y. Hussain. (2020). Network Intrusion Detection for Distributed Denial of- Service (DDoS) Attacks Using Machine Learning Classi cation Tech niques. [Online].Available: http://hdl.handle.net/1828/11679

[14] J. P. A. Maranhão, J. P. C. L. da Costa, E. P. de Freitas, E. Javidi, and R. T. de Sousa Júnior, Error-robust distributed denial of service attack detection based on an average common feature extraction technique, Sensors, vol. 20, no. 20, p. 5845, Oct. 2020, doi: 10.3390/s20205845. [19] C.-S. Shieh, W.-W. Lin, T.-T. Nguyen, C.-H. Chen, M.-F. Horng, and D. Miu, Detection of unknown DDoS attacks with deep learning and Gaussian mixture model, Appl. Sci., vol. 11, no. 11, p. 5213, Jun. 2021, doi: 10.3390/app11115213.

[15] S.SindianandS.Sindian, Anenhanceddeepautoencoder-basedapproach for DDoS attack detection, WSEAS Trans. Syst. Control, vol. 15, pp. 716724, Dec. 2020,doi: 10.37394/23203.2020.15.72.

[16] R. Al-Saadi, G. Armitage, J. But, and P. Branch, A survey of delay based and hybrid TCP congestion control algorithms, IEEE Commun. Surveys Tuts., vol. 21, no. 4, pp. 36093638, 4th Quart., 2019, doi: 10.1109/COMST.2019.2904994.

[17] B. Sun, L. Yang, W. Zhang, M. Lin, P. Dong, C. Young, and J. Dong, SuperTML: Two-dimensional word embedding for the precog nition on structured tabular data, in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2019, pp. 19, doi: 10.1109/CVPRW.2019.00360.

[18] G. Ajay and P. Krishnan. (2018). A Study and Analysis of Effective Data Transmission Using UDP. [Online]. Available: https://www. ijser.org/researchpaper/A-Study-and-Analysis-of-Effective-Data transmission-Using-UDP.pdf

[19] K. Alieyan, M. M. Kadhum, M. Anbar, S. U. Rehman, and N. K. A. Alajmi, An overview of DDoS attacks based on DNS, in Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC), Oct. 2016, pp. 276280, doi: 10.1109/ICTC.2016.7763485.

[20] L.Breiman, Usingiterated bagging to debias regressions, Mach. Learn., vol. 45, pp. 261277, Dec. 2001, doi: 10.1023/a:1017934522171. [26] K. Q. Weinberger, J. Blitzer, and L. Saul, Distance metric learning for large margin nearest neighbor classification, J.Mach.Learn.Res.,vol.10, pp. 207244, Jul.2009, doi:10.5555/1577069.1577078.

[21] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu, LightGBM: A highly ef cient gradient boosting decision tree, in Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS), 2017, pp. 31493157, doi: 10.5555/3294996.3295074.

[22] T. Chen and C. Guestrin, XGBoost: A scalable tree boosting system, in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2016, pp. 785794, doi: 10.1145/2939672.2939785.

[23] R. Schapire, Explaining AdaBoost, in Empirical Inference. Berlin, Germany: Springer, 2013, pp. 3752, doi: 10.1007/978-3-642-41136-6_5.

[24] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, Linear discriminant analysis: A detailed tutorial, AI Commun., vol. 30, no. 2, pp. 169190, 2017, doi: 10.3233/AIC- 170729.

[25] A. Mansour, Texture classi cation using Naïve Bayes classi er, Int. J. Comput. Sci. Netw. Secur., vol. 18, no. 1, pp. 19, 2018. [Online]. Available: http://paper.ijcsns.org/07_book/201801/20180113.pdf [32] T. Evgeniou and M. Pontil, Support vector machines: Theory and applications, in Machine Learning andits Applications, vol. 2049. Berlin, Germany: Springer, 2001, pp. 249257, doi: 10.1007/3- 540-44673-7_12.

[25] J. Halladay, D. Cullen, and N. Briner. DDoS Time-based 881 Experimentation, Github Repository, Jan. 2022. [Online]. Available: https://github.com/jehalladay/DDoS_Research

DOI: 10.48175/IJARSCT-22158