International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

The Investigative Study on the Performance Analysis of SMOTE employed Machine Learning Classifier Models to DDoS Attack Detection

[1] Sravan Kumar G, [2] Dr. M Sunitha, [3] Ghantasala Srinivasa Rithik, [4] S Veeresh Kumar, [5] Dr K Sreerama Murthy
[1] CVR College of Engineering, Hyderabad, [2] CVR College of Engineering, Hyderabad, [3] CVR College of Engineering, Hyderabad, [4] St. Martin's Engineering College, Secunderabad, [5] Koneru Lakshmaiah Education Foundation, Hyderabad.
[1]sravankumarcvr@gmail.com , [2] palemonisunitha@gmail.com, [3] srithik2002@gmail.com, [4]salvadiveeresh2023@gmail.com, [5]drsreeram1203@gmail.com

Abstract— Distributed Denial of Service (DDoS) attack, a severe attack on the network services during the contemporary era, is categorized under active attacks in security attacks. The impact of this attack on the organization or individual resources leads to massive loss in terms of finance, reputation. Therefore, detecting Distributed DDoS attacks is vital in ensuring the availability and integrity of online services of an organization. The work in this paper employed machine learning techniques, complemented by Synthetic Minority Over-sampling Technique (SMOTE), to tackle the inherent challenge of imbalanced DDoS attack dataset: CSE-CIC-2018 and to enhance computational efficiency while maintaining accuracy with a fraction of the original dataset. The emphasis of the this works is to comprehensively assess the performance of five prominent algorithms of machine learning - Naive Bayes, Random Forest, Logistic Regression, Decision Tree, and XGBoost - in the context of detection of DDoS attack. The overhead of oversampling is handled with the application of SMOTE oversampling and it has been addressed data imbalance issues, improving the algorithms' capability to identify attacks of DDoS effectively. The work of this paper finds and reveals distinct comparative advantages among the algorithms employed in the DDoS attack detection and provides actionable insights in choosing the most suitable algorithms of Machine learning for the detection of DDoS attack, provided emphasizing the significance of SMOTE to enhance the algorithms' performance in the presence of imbalanced data. Eventually, this paper offers invaluable guidance for organizations seeking to make safe their network against DDoS attacks while considering the crucial tradeoffs between accuracy and computational efficiency. The proposed work in this paper presented the results that Random Forest classifier ensured the better performance with F1-Score value 0.99, Mathews Correlation Coefficient (MCC) value 0.98 and accuracy value 0.99 relative to other classifiers employed.

Keywords- Distributed Denial of Service, DDoS attack, Machine Learning, SMOTE, Naïve Bayes, Random Forest, Logistic Regression, Decision Tree, XGBoost, Mathews Correlation Coefficient, F1-Score.

I. INTRODUCTION

In terms of resources and time, an active attack has vital implications for IT infrastructure. Cyber attacks that include attack of DDoS causes financial losses to organizations and businesses. The most hazardous attack is DDoS attacks and it has been produced literature in this area [11, 12, 13]. These attacks directly affect the economic and financial sector. For example, the Mirai attack in October, was a series of DDOS attacks targeting the "DNS"; supplier Dyn and its operations on October 21, 2016 [11, 12, 13]. Attacks of these kind resulted service interruption of platforms of the Internet over multiple regions across North American and European nations [11, 12, 13]. The first DDoS attack that cut off all Internet access in a city for several hours occurred in 1997 at the hacker conference in Las Vegas by attacker Khan C Smith [14]. After this attack, many online attacks took place against Sprint, EarthLink, E-Trade and many popular Internet service companies [14]. In 2001, Smith created the first botnet that used fake domains, email addresses and websites to spam nearly a quarter of all spam on the Internet [14]. The DDoS attack on github was one of the destructive attacks that happened with 1.3 Tbps of incoming traffic and transfer rate about 126.9 million bits/sec [15]. An open source software system called memcached, used to accelerate networks and web services, has been hacked [16]. An attacker spoofs requests to a vulnerable server by overloading github with Internet traffic [16]. Due to overload, Internet resources and infrastructure cannot handle any requests, leading to denial of service [16]. Attackers influence Memcached's amplification effect by a factor of 50,000 by flooding it with fake requests [16]. The motivation of DDoS attacks is diverse and falls under the categories such as Ideology, Business Competitors, Cyber Warfare, Extortion and Boredom [16]. DDoS attacks have been classified in to the following: attacks of Volume based, attacks of Protocol based, and attacks of Application based [16]. In this paper, author presents DDoS attack of flood types: UDP, HTTP, SYN. NTP, Zero Day attack and worked on the classification of these attacks by using techniques such as Gaussian Naïve Bayes, The Stochastic Gradient Descent (SGD), Random Forest, Support

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

Vector Machine, K-Nearest Neighbours [1]. There are two types of DDoS defense attacks: defense of source and defense of destination [5]. The defense from target-side is the victim-side protection to disconnect the connection as soon as attack of DDoS gets detected, whereas the defense of source identify the attack of DDoS by the analyzing traffic from the source and traffic landing page assumes that incoming and outgoing traffic are proportional to each other [5]. Classification that is used to detect DDoS attacks using IP address, destination port and flow density, requires algorithms from machine learning that include Random Forest algorithm, naive Bayes, and support vector machine are employed [17]. The attack of DDoS detection method has been carried out by using two steps: feature extraction and pattern detection which are part of machine learning [17].

II. RELATED WORK

DDoS Attacks are inevitable to detect with one appropriate and generalized machine learning algorithms and the authors have investigated popular machine learning methods on the CICDoS2019 dataset with the direction that the hybrid algorithms to be tested for better performance in the future work [1]. In this work, authors have identified the DDoS attack, specifically the Ping of Death attack by the Random Forest with accuracy value equal to 0.998 [2]. In this work, Splunk software collected samples of data packets where there exist both normal and attacked samples [2]. The authors in this paper proposed methods of DDoS attack through detection and mitigation of traffic coming from the BOTNET to the server [3]. The literature of this work offered methodology based on machine learning to analyze the DDoS traffic and detect the attack [3]. This paper highlighted the various literature works on DDoS attack detection and provided detailed analysis of various algorithms like Random Forest and CNN employed on various datasets [4]. The work presented in this paper presented detailed study of machine learning algorithms on the datasets available includes NSL-KDD, ICDX, CIDDS-001, CICIDS 2017 in the cloud environment [5]. The author presented the work to have deeper understanding of the issues of attacks of DDoS and developed classification defense systems appropriately detect attacks of DDoS [6]. The state of art of the literature proposed novel method where traces in the traffic flow has been classified as normal and abnormal traffic traces using Naïve Bayes Algorithm and Random Forest among which Naïve Bayes yielded better performance related metric values relative to Random Forest Algorithm[7]. In this work authors proposed the approach to detect and classify category of attacks include flood of HTTP, SID DoS and Traffic obtained normally using WEKA and highlighted that the alogorithm-J48 presented best results relative to Random Forest and Naïve Bayes [8]. This paper emphasized the development and design the system that identifies attack of DDoS and prevents referred as CloudGuard a cloud-based system which employed analysis based on volume and statistical web-profile statistic based approach [9]. The authors presented detection of DDoS through the model to detect effectively and provide appropriate response to attacks of DDoS [9]. The authors emphasized on the identification of HTTP attacks of DDoS on the environment of cloud through the attack identification system that employs Entropy of Information Theory and Ensemble Learning of Random Forest [10]. The network header's incoming traffic entropy is calculated through using sliding window algorithm which is of time-based [10]. It has been conducted the experiment on CIDDS-001 Classification task which gets triggered when the calculated entropy goes beyond its usual range on the public dataset CIDDS-001 [10].

III. METHODOLOGY

A. CSE-CIC-IDS2018 Dataset

The dataset was originally created by the University of New Brunswick for studying DDoS data [18]. This dataset has been obtained fully from 2018 and the dataset itself has been based on logs of the servers of the university, where different DoS attacks have been detected and made this dataset publicly available [18]. The CSE-CIC-IDS2018 dataset shows diverse formats of attacks from the University of New Brunswick [18]. Totally, eighty attributes exist in the dataset and IDS logging system that maintains each entry has been installed in University of New Brunswick [18]. The significant attributes of the dataset include: Dst_Port (Destination port), Protocol, flow_duration, tot_fwd_Pkts, tot_bwd_pkts , label [18].

B. Data Pre-processing

The reliability besides accuracy of models of machine learning is relied upon effective pre-processing of the raw dataset. In this work, identifying and handling the null values present in the dataset is one of the challenging and fundamental steps of the data pre-processing. This fundamental step involves identifying the columns with missing data and deciding upon appropriate strategies, such as imputation or removal, ensures that the dataset remains robust and accurate. Infinite or excessively large values can distort analysis and impact model performance. The proposed work ensured the second pre-processing step that involves identifying and eliminating such values to prevent unwanted model biases and anomalies.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

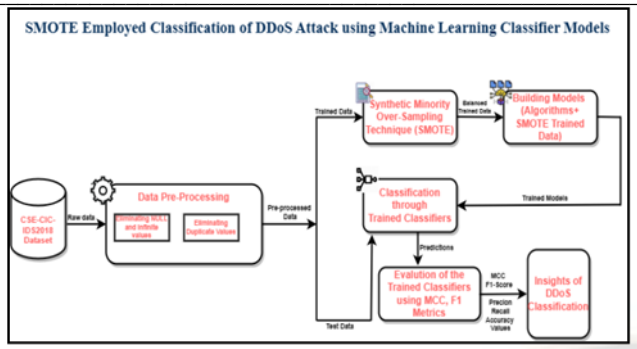

Figure 1: Methodology of the work

In this work, it is identified situation where object data type results setback to the machine learning algorithm performance. Therefore, the data type of features, a crucial consideration requires converting these features into suitable numeric or categorical data types using tools like the **.astype()** method in Pandas facilitates subsequent analysis. The proposed work emphasized on Duplicated entries that these can introduce bias and skew results, so identifying and removing them is considered as the fourth step. This enhances the dataset's integrity and reliability, ensuring that each data point is unique and contributes meaningfully to the analysis. The fourth step, identifying and removing the duplicate entries results in the elimination of bias and skewness in the dataset. This step in the proposed work enhanced the dataset's integrity and reliability, ensuring that each data point is unique and contributes meaningfully to the analysis. Sub sampling the fraction of the dataset provides practical solution for the initial analysis and model development in the case where large dataset is exceptionally large and exceeds computational capabilities. The proposed work is focused on the classification of the malicious activities in the dataset and the dataset is prepared for binary classification problem with categorical label Benign represented as 0 and Malicious (Malign) represented as 1. Eventually, features have been applied to Min-Max scaling and normalized the appropriate features within 0-1 range to improve model compatibility and improved convergence. This scaling in the proposed work simplified the comparison and interaction of features and thus promoting machine learning models to be more effective.

C. Training the Models

The proposed work trained the machine learning algorithms XGBoost and algorithms employed in this work harnessed these trained algorithms capabilities for DDoS attack detection by leveraging the CSE-CIC-2018 dataset. The training process of this work is initiated by separating the dataset into training and testing sets, adhering to standard machine learning practices. The test dataset comprises of 20% of the total dataset is adopted for being evaluating performance of the trained classifiers. In this work, to address the imbalanced data distribution of the dataset, Synthetic Minority Over-sampling Technique (SMOTE) sampling method has been employed to ensure that classifiers are trained on an imbalanced dataset, mitigating potential biases toward the majority class and enhancing its effectiveness in detecting DDoS attacks.

Logistic Regression: The proposed work employed logistic regression which is fed with appropriately prepared dataset obtained from the data pre-processing step. The trained logistic regression classifier which is fitted on the resampled training data employed the following hyper parameters: max_iter and random_state. The max_iter=1000 hyperparameter is primarily adjusted to ensure that the logistic regression algorithm converges to a solution. the random_state, make the experiments and results more transparent and easier to validate.

Gaussian Naïve Bayes (GNB): The GNB classifier which does not have a prior's parameter uses default class priors which ensure class priors probability are estimated from the training data. This approach is appropriate for research purposes as it ensures that the model's priors are representative of the training data. GNB classifier includes a hyper parameter var_smoothing which represents a small, positive value added to the variances of all features. This is used to prevent zero variances, which can cause issues in probability calculations. By default, var_smoothing has set to $1e^{-9}$ that represents very little value that helps avoid division by zero. This default value is suitable for many research scenarios, but it can be adjusted if you encounter numerical stability issues. The GNB classifier assumes that the features are normally distributed within each class. The above mentioned hyper parameters mentioned above in the GNB ensured the appropriate values to create GNB which has been trained on the dataset.

Decision Tree: The classifier in the work employed the hyper parameters such as max_depth., min_samples_split., min_samples_leaf., max_features. and criterion. The hyper parameter max_depth limits the depth value to avoid over fitting and smaller value of max_depth creates shallow tree that captures noise in the data. The hyper parameter min_samples_split., controls the samples that are minimum and required to split an internal node. However, it's important to balance this value based on your dataset's size and complexity. The third hyper parameter controls samples that are minimum required to be in a leaf node and also ensure model generalization through the prevention in the creation of very small leaf nodes that capture noise. The problem of over fitting is prevented with promotion of diversity in feature selection for different branches of tree and sqrt algorithm that calculates the value obtained when total number of features undergoes square root, decides no. of feature to consider for each split. This work employed gini criterion that determines

how often a randomly chosen element would be incorrectly classified. Decision tree's hyper parameters have been chosen in this work appropriately to build a classifier-decision tree which is relatively shallow and less prone to over fitting.

Random Forest Classifier: In this proposed work, employed classifier Random Forest used the hyper parameters class_weight which is set to balance. This is particularly used when the dataset is imbalanced and also this hyper parameter mitigates the impact of class imbalance by assigning higher weights to the minority class. max_depth., similar to classifier Decision Tree, limited the depth of the trees which is maximum and its value is within the Random Forest is set to 3. This constraint prevents individual decision trees in the ensemble from becoming overly complex and over fitting the data. min_samples_split., and min_samples_leaf., values are set to 5 to these hyper parameters and imposed constraints on samples that are minimum, required to split an internal node and also represents leaf node's minimum number of samples. These values (5 in your case) ensure that the ensemble decision tree will not make overly fine-grained splits, which helps prevent over fitting. In this work, these values are practical choice because it promotes a balance between model complexity and generalization. max_features='sqrt' hyper parameter limits the count of features considered for each split to the value obtained from the square root function that takes all features is a common heuristic in Random Forests. It introduces randomness and decorrelates the individual trees, making the ensemble more robust and less prone to over fitting. This choice is suitable for this work because it helps to create a diverse set of decision trees. The hyper parameter, n_estimators=50, represents the forest's that contain tree count. A larger value of trees in the forest generally improves the performance of the Random Forest ensemble. However, the choice of 50 is reasonable as it strikes a balance between computational cost and performance. It can capture patterns in the data while remaining tractable for the proposed work. The hyper parameter criterion='gini'that has been chosen the Gini impurity as the impurity criterion is a common choice suitable for classification problems and this Gini impurity measures the frequency of misclassifications and is effective for this proposed work.

XGBoost: The one of the hyper parameters chosen for the XGBoost Classifier in this work is objective="binary:logistic". The objective hyperparameter specifies the learning task and the corresponding objective function. In this case, "binary:logistic" is chosen, which indicates that the model is trained for binary classification using logistic regression as the objective function. This is a suitable choice for many binary classification problems. eval_metric="logloss": The eval_metric hyperparameter in the XGBoost determines the evaluation metric to monitor during training. "logloss" (logarithmic loss) is a widely employed metric for binary classification tasks. It determines the predicted probabilities and the true labels, dissimilarity. By minimizing the logloss, the model aims to provide well-calibrated probability estimates, which is important in many research scenarios. max_depth , the one of the hyper parameters in the XGBoost, set to 3 represents the depth of the each tree which is maximum in the XGBoost to 3. Limiting tree depth helps prevent over fitting, as shallow trees are less likely to capture noise in the data. This choice balances model complexity and generalization, making the model more interpretable and less prone to over fitting. The hyper parameter learning_rate set to 0.1 in this work controls the size of the step at the every iteration with intent of approaching toward a very less value of the function that represents log loss. A 0.1 value is a reasonable starting point and is often used for gradient boosting. It's large enough to allow the model to converge relatively quickly, yet small enough to prevent overshooting the optimal solution. The learning rate can be fine-tuned based on the specific dataset and experimentation. The hyper parameter n_estimators sets value that accounts boosting rounds or trees in the ensemble. In this work it is chosen 100 trees, which maintains balance between complexity of the model and efficiency of the computation. The performance gets enhanced with increase in number of trees, but it comes at a cost of longer training times. Setting the hyper parameter random_state to 42 ensures reproducibility of the experiments. By using the same random seed (42 in this case), it is replicated the results in subsequent runs. This is crucial, as it allows you to maintain consistency and facilitate result verification.

D. Evaluating the Models through Performance Metrics

In a context of classifying imbalanced data with the use of SMOTE (Synthetic Minority Over-Sampling Technique.), performance metrics like Mathews Correlation Coefficient, F1-Score, recall, precision, and accuracy play crucial roles in assessing the effectiveness of the classification model. The model's confusion matrix presents a detailed representation of the predictions, including true positives (correctly identified minority class instances), true negatives (correctly identified majority class instances), false positives (majority class instances incorrectly classified as minority), and false negatives (minority class instances incorrectly classified as majority) [19]. In imbalanced datasets, understanding these specific prediction outcomes is vital for evaluating the performance of the model.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

Table 1: Trained Classifier Models with the corresponding performance metric values

	Accuracy	Precision	F1-Score	Recall	Matthews Correlation Coefficient
Logistic Regression	0.97	0.98	0.97	0.97	0.94
Gaussian Naïve Bayes	0.76	0.87	0.78	0.76	0.6
Decision Tree Classifier	0.98	0.99	0.98	0.98	0.91
Random Forest Classifier	0.99	0.99	0.99	0.99	0.98
XGBoost Classifier	0.88	0.92	0.88	0.88	0.76

Metrics which serve the specific purpose in evaluating how well the model handles imbalanced datasets are Mathews Correlation Coefficient, accuracy, precision, recall, F1 Score and ROC-AUC [19]. In the work that is proposed, the trained models performance on the dataset CSE-CIC-2018 is evaluated with performance metrics.

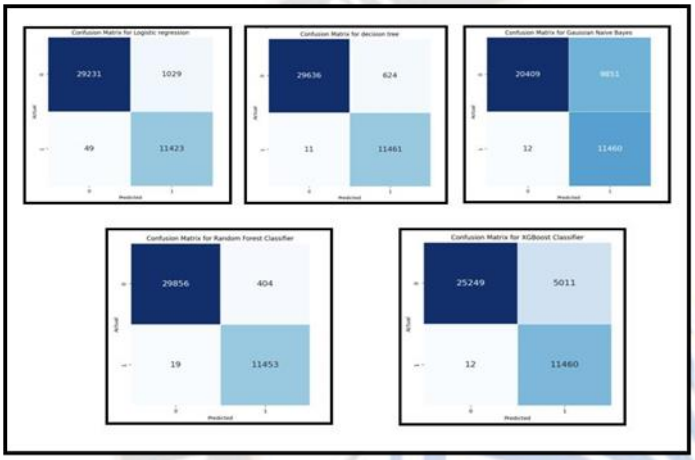

Figure 2: Confusion Matrices of the Trained Classifier Models

IV. RESULTS AND DISCUSSIONS

This work revealed that classifier Random Forest provided relatively better results to other employed trained models of the work as shown in the figure 2, in terms of metrics of the performance include Mathews Correlation Coefficient, accuracy, precision, recall and F1-Score. An ensemble method of Random Forest, merge multiple decision trees to have results in improved generalization and enhanced performance relative to individual decision trees. The problem of over fitting and capturing of complex patterns in the data is achieved through the combination of multiple trees. The Random Forest ensures performance with improved ability to handle imbalanced class. The hyper parameter class_weight is to 'balanced' ensures class weights in such a way that significance has been given to minority class. Eventually, Random Forest avoids over fitting, robust to noisy or inappropriate features, has the ability to handle datasets which is again combination of appropriate and inappropriate features.

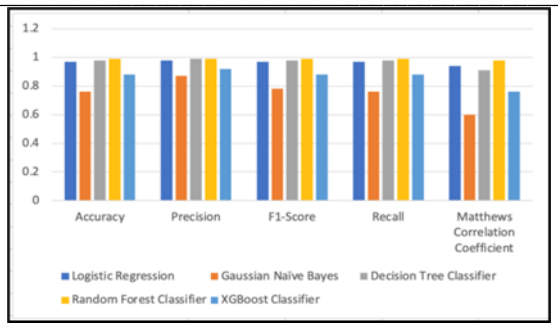

Figure 3: Bar Graph Analysis of Trained Classifier Models with the performance metric values

MCC is one of the metric that considers True Negatives which impacts the performance of the metric when positive class is inverted as Negative Class and Negative Class is inverted as Positive Class. In this regard, F1- Score offers less performance, hence then MCC metric is used for the evaluation of the performance of trained classifiers with respect to the dataset employed in this work. Random Forest outperformed remaining classifiers employed in this work and it has been shown in the figure 4.

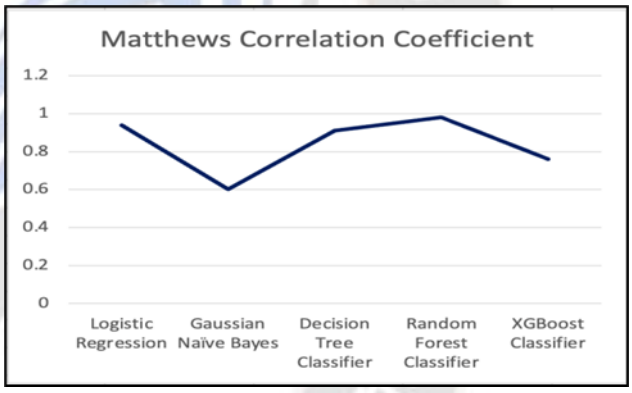

Figure 4 Line Graph Analysis of Trained Classifier Models with the MCC performance metric values

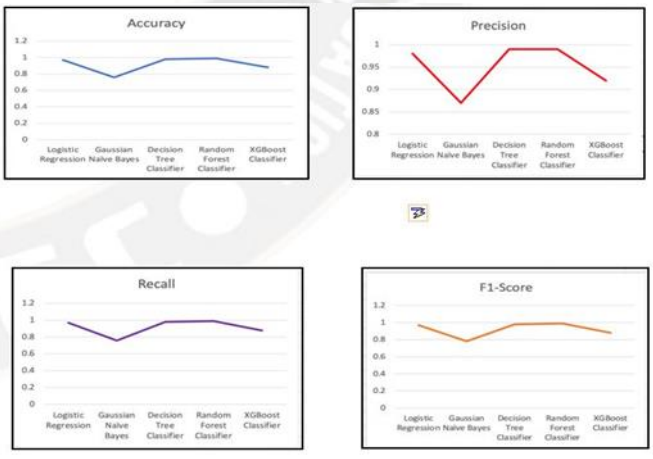

Figure 5: Line Graph Analysis of Trained Classifier Models with the each performance metric values

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

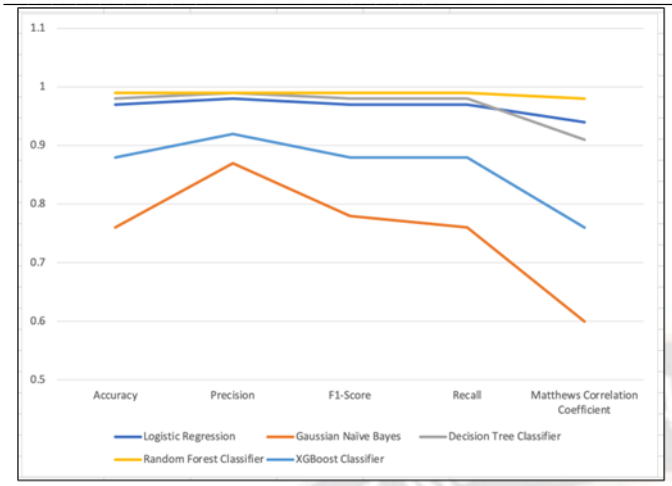

Figure 6: Line Graph Analysis of Trained Classifier Models with the performance metric values

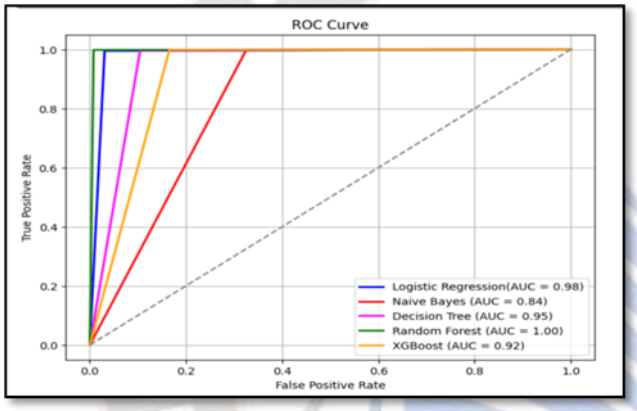

Figure 7: ROC-AUC Analysis of Trained Classifier Models with the corresponding performance metric values

Logistic Regression classifier, good at linear data, struggled to the dataset in this work due to imbalanced dataset and noisy data makes classifier inaccurate. Gaussian Naïve Bayes classifier in this work failed to draw complex decision boundaries for the employed dataset in this work. This work proved that decision tree provided relatively lower performance to the classifier random forest due to default hyper parameters employed in decision tree classifier. XGBoost, a powerful gradient boosting algorithm, offers better performance, if the hyper parameters are tuned appropriately with respect to the dataset. XGBoost yields sub optimal results if the hyper parameters to this classifier are not set optimally according to the dataset.

CONCLUSION & FUTURE WORK

The DDoS attack detection in this work, ensures services of an organization with higher availability and reliability, has been given greater significance to binary classification of the access to the services of an organization. In this work access to the services is classified into two types of access which are adversarial access, malign and authorized benign access. The imbalanced dataset employed in this work has been transformed into balanced dataset through the SMOTE and fed that balanced dataset to machine learning algorithms with fine tuned hyper parameters with respect to the tradeoffs involved in the respective machine learning algorithms. In this work, it has been normalized the labels of the dataset to benign access and malign access, thus leads to binary classification. The trained machine learning classifier models to DDoS attack detection offered better results in classifying the malign and benign access. The classifier models in this work are Random Forest, Logistic Regression, Gaussian Naïve Bayes, Decision Tree and XGBoost are trained and evaluated with performance metrics such as accuracy, precision, recall, F1-score and ROC-AUC curve. Eventually, the proposed work performed detailed analysis on the performance of the classifier models and revealed the results that Random Forest classifier with accuracy 0.99 and F1-score 0.99 outperformed the other classifier models employed in this work. The future work directs the DDoS attack detection enhancement with the integration of transfer leaning for the improved adaptability. The future contributions to the proposed work such as exploration of feature engineering and employing hybrid models can lead to the robust and effective DDoS attack system against the adversarial access and thus enable this system to be practically deployed in the real-time networks.

REFERENCES

[1] R. S. Devi, R. Bharathi and P. K. Kumar, "Investigation on Efficient Machine Learning Algorithm for DDoS Attack Detection," *2023 International Conference on Computer, Electrical & Communication Engineering (ICCECE)*, Kolkata, India, 2023, pp. 1-5, doi: 10.1109/ICCECE51049.2023.10085248.

[2] M. C, K. K. B, T. Kumar A, V. B and H. K. V, "Detection of Distributed Denial of Service Attack using Random Forest Algorithm," *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)*, Pudukkottai, India, 2022, pp. 382-386, doi: 10.1109/ICACRS55517.2022.10029249.

[3] D. Satyanarayana and A. S. Alasmi, "Detection and Mitigation of DDOS based Attacks using Machine Learning Algorithm," *2022 International Conference on Cyber Resilience (ICCR)*, Dubai, United Arab Emirates, 2022, pp. 1-5, doi: 10.1109/ICCR56254.2022.9995773.

[4] S. Singh, M. Gupta and D. K. Sharma, "DDOS Attack Detection with Machine Learning: A Systematic Mapping of Literature," *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2023, pp. 939-945, doi: 10.1109/ICSSIT55814.2023.10060897.

[5] C. Sathvika, V. Satwika, Y. Sruthi, M. Geethika, S. Bulla and S. K, "DDoS Attack Detection on Cloud Computing Services using Algorithms of Machine Learning: Survey," *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2023, pp. 1094-1100, doi: 10.1109/ICCMC56507.2023.10083549.

[6] C. Douligeris and A. Mitrokotsa, "DDoS attacks and defense mechanisms: a classification," *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology (IEEE Cat. No.03EX795)*, Darmstadt, Germany, 2003, pp. 190-193, doi: 10.1109/ISSPIT.2003.1341092.

[7] G. Ajeetha and G. Madhu Priya, "Machine Learning Based DDoS Attack Detection," 2019 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 2019, pp. 1-5, doi: 10.1109/i-PACT44901.2019.8959961.

[8] P. S. Saini, S. Behal and S. Bhatia, "Detection of DDoS Attacks using Machine Learning Algorithms," 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2020, pp. 16-21, doi: 10.23919/INDIACom49435.2020.9083716.

[9] P. S. Saini, S. Behal and S. Bhatia, "Detection of DDoS Attacks using Machine Learning Algorithms," 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2020, pp. 16-21, doi: 10.23919/INDIACom49435.2020.9083716.

[10] Mohamed Idhammad, Karim Afdel, Mustapha Belouch, "Detection System of HTTP DDoS Attacks in a Cloud Environment Based on Information Theoretic Entropy and Random Forest", 2018 Hindawi, Security and Communication Networks, Volume 2018, Article ID 1263123, 13 pages, https://doi.org/10.1155/2018/1263123

[11] The 2016 Dyn Attack and its Lessons for IoT Security | MS&E 238 Blog (stanford.edu)

[12] H. Sinanović and S. Mrdovic, "Analysis of Mirai malicious software," 2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2017, pp. 1-5, doi: 10.23919/SOFTCOM.2017.8115504.

[13] Yamaguchi, Shingo and Gupta B B, "Malware Threat in Internet of Things and Its Mitigation Analysis', Security, Privacy, and Forensics Issues in Big Data, 2020, 10.4018/978-1-5225-9742-1.ch016.

[14] Smith, J. (2019). 2019 Year of DDoS?. Hostdime blog. https://www.hostdime.com/blog/2019-ddos-protection. (accessed on April 13, 2020)

[15] Cloudflare Inc, USA. Famous DDoS Attacks. The Largest DDoS attacks of all time, https://www.cloudflare.com/ learning/ddos/famous-ddos-attacks. (accessed on April 13, 2020)

[16] U. Rahamathullaha , Dr. E. Karthikeyanb, "Distributed denial of service attacks prevention, detection and mitigation – A review", International Conference on Smart Data Intelligence (ICSMDI 2021)

[17] Jiangtao Pei, Yunli Chen and Wei Ji, "A DDoS Attack Detection Method Based on Machine Learning," Journal of Physics: Conference Series, Volume 1237, Pages: 32-40, July 2019

[18] IDS 2018 Intrusion CSVs (CSE-CIC-IDS2018), IDS 2018 Intrusion CSVs (CSE-CIC-IDS2018) (kaggle.com).

[19] M. . Sunitha, K. . Manasa, S. . Kumar G, B. . Vijitha, and S. . Farhana, "Ascertaining Along With Taxonomy of Vegetation Folio Ailment Employing CNN besides LVQ Algorithm", IJRITCC, vol. 11, no. 6, pp. 113–117, Jul. 2023. https://doi.org/10.17762/ijritcc.v11i6.7278