Interpretable DDoS Attack Detection: Combining Machine Learning and SHAP

***Abstract*—In today's world, technology has significantly advanced across all sectors of life. As computers are becoming smaller, faster and more accessible, it also presents a significant challenge in maintaining network security to protect private information and ensure the reliability of networks. It has become a priority for technology experts, particularly in defending against cyberattacks. Therefore, this study focuses on detecting Distributed Denial of Service (DDoS) attacks, specifically those that can target many servers and web applications. In this research, a new taxonomy was developed for classifying the attack into two categories, Reflection based attack and Exploitation based attack to enhance detection accuracy and better performance of the model. Several machine learning models, such as Random Forest, Naive Bayes, Decision Tree, and XGBoost, have also been implemented on the CIC-DDoS2019 dataset. Besides, Explainable AI (SHAP) technique has been introduced for models performance interpretation. The results demonstrated a high level of accuracy, achieving 99.89% for exploitation-based attacks and 99.74% for reflection-based attacks, showing substantial improvement in detection rates while minimizing processing time.**

***Index Terms*—DDoS Attack, Cyber Security, Machine Learning, Random Forest Classifier, Explainable AI.**

I. Introduction

Cyber security is a significant concern nowadays, particularly with the growing reliance on the IT sector of every sector of life. As cyber threats become more sophisticated, the necessity for ongoing innovation and investment in cybersecurity strategies continues to grow. One of the most common threats in the cyber world is Distributed Denial of Service attack (DDoS), where a malicious entity floods a server or a website with excessive data which is caused by the attackers to disrupt its operation. This results in service interruptions for the legitimate users. Also, it can end up by crashing the server or website. Unlike Denial of Service attacks, which originate from a single source, DDoS attacks are generated from multiple sources, which are often coordinated through botnets. It is easier for the attackers to cause the attack by setting up a botnet from their end. To detect the threats, numerous studies have introduced new taxonomies emphasizing the importance of recognizing the source of that attack. Most of the researchers have conducted their study using ML and DL classifiers. On the other hand, attackers are also now leveraging novel strategies, such as exploiting TCP/UDP-based application-layer protocols [1], to enhance the severity of their attacks. As network defense resources become more limited, mitigating DDoS attacks become increasingly challenging for the experts. Differences in network capacity, scalability, and mitigation options across infrastructures add complexity. Key decisions, such as choosing the number of security filters or between Flowspec and NETCONF for routing, influence the effectiveness of defense strategies. However, with AI and machine learning advancements, DDoS activity can now be detected early, enabling more targeted and efficient mitigation. Incorporating big data analytics and AI/ML into a comprehensive DDoS defense strategy helps organizations proactively identify and neutralize threats, ensuring network security and uninterrupted service. The main contributions of this study are

- Analyzing the data and dividing them into two types of Attacks such as Reflection based and Exploitation based.
  Exploitation-Based Attacks: In these attacks, attackers use the victim's IP to send reply packets through reflection servers. Protocols like TCP and UDP are exploited, with SYN and UDP flood attacks overwhelming the target through numerous packets [2].
  Reflection-Based Attacks: These attacks focus on concealing the attacker's identity by manipulating internet services, such as DNS, NTP, and SNMP servers, which lack comprehensive logs, making it harder to trace the attacker's origin. This tactic is a major reason for the popularity of DDoS strategies [2].
- Some ensemble classifiers such as Random Forest (Bagging), XGBoost (Boosting) and Decision Tree have been proposed in this research. Also, a baseline classifier like Gaussian Naive Bayes has also been implemented in the work.
- Extra Tree Classifiers have been performed to select the best features. Also, a batch size has been taken to reduce the time consumption.
- SHAP method is introduced for models' performance visual interpretation.

II. Literature Review

DDoS detection is a large data issue with relatively unrestricted resources, limited only by the processing platforms. However, DDoS mitigation is a challenge when resources are limited. So some of the existing works and research projects have been analyzed for this work.

Saini et al. [2] worked in the area of network security, where they considered DDoS as the most harmful attack. According to them, systems continue to be overloaded with the continuous fake requests of BOTs rather than provide services to real users at the attacking phase. The researchers employed a machine learning-based tool to identify and categorize various network traffic flow categories in the study. They used the CIC-DDoS2019 dataset that contains a mix of many contemporary

sorts of attacks, including HTTP flood, SID DoS, and regular traffic, which is used to validate their suggested approach. Their research shows that the J48 algorithm gave the best outcome compared to the Random Forest model and Naive Bayes model.

Dasari et al. [1] stated in their paper that the security systems of computer networks and information technology are targeted for disruption by the severe DDoS attack. They took Syn flood, UDP flood and UDP-Lag datasets from CI-CDDoS2019 for their research and customized the data. They found uncorrelated feature subsets in the datasets and followed Pearson, Spearman and Kendall correlation approaches. Then they chose the common features from their customized feature subsets. For the experimentation, they applied basic classifiers like Logistic Regression, Decision Tree, KNN, Naive Bayes, Random Forest. Also, Ada Boost, Gradient Boost and Multi Layer Perceptron (MLP) were used for the classification process. Finally, the accuracy, precision, recall, F1-score, specificity, log loss, execution time, and K-fold cross-validation of various classification parameters are generated to determine the accuracy of detection.

Al-Shareeda et al. [3] considered DDoS attack as a great threat of network security in their recent work. According to them, the attack aims to shut down the central server by overloading it with numerous requests until it reaches its capacity. Also, the attack is very risky as it doesn't need a lot of effort or specialized equipment. Mainly, it relies on a large number of bots that are controlled by a single master bot holding a false IP address. The master bot is actually controlled by the attacker. In the study, the authors have examined several machine learning (ML) and deep learning (DL) based methods for detecting and assessing the attacks.

Aktar et al. [4] proposed a deep learning approach on the same dataset to detect the attack. They trained the model for learning the normal traffic pattern from the train data. They built a Contractive Autoencoder for their deep learning model and got a detection accuracy of the attack between 93.41% and 97.58%.

Kumar et al. [5] implemented another deep learning based model LSTM (Long Short-Term Memory) on the same data. So they also performed feature selection and extraction process. After the train and test process, they claimed that they can achieve a detection accuracy of 98%.

Sayed et al. [6] presented a multi-classifier model on the same dataset using stacking ensemble deep neural networks to identify various types of DDoS attacks. They proposed Convolution Neural Network (CNN), Long Short Term Memory (LSTM), and Gated Recurrent Unit (GRU) models in their work. When evaluating models with huge datasets, the ensemble strategy improves model performance. The suggested model outperforms other comparable methods, and the proposed model achieves an accuracy of 89.4%, outperforming other similar methods.

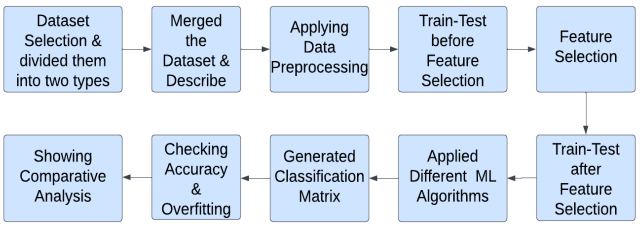

Fig. 1. Workflow of proposed methodology

III. Proposed Methodology

The top-level overview of the research is shown in the Fig-1. The implementations started with the combined datasets and then data preprocessing and the Machine Learning classifiers have been implemented.

A. Dataset

In this research, the CIC-DDoS2019 dataset served as the main source which integrated some single csv file to produce a new dataset. Many recent attack data that closely resembles real-world scenarios as well as benign traffic have been found in the dataset. There are more than ten separate csv files of data that represents various packet types such as SYN, UDP, UDPLag, ICMP, LDAP, NTP, DNS, SNMP, PortMap, NetBIOS and MSSQL in the CIC-DDoS2019 collection [7]. For this study, six single datasets were selected which are Syn,csv, UDPLag.csv, DrDoS-UDP.csv, DrDoS-NTP.csv, DrDoS-DNS.csv and DrDoS-LDAP. While numerous surveys have suggested taxonomies for detecting the attack working on the same datasets, the aim of this research was to find new attacks and develop a method to differentiate between the attacks which are Reflection-based and those which are Exploitation-based. Our analysis explored potential new attacks involving HTTP and TCP/UDP based protocols in the application layer. The resulting datasets comprised two csv files which contain three types of attack under each type. To be more precise, the Reflection based category has DrDoS-DNS, DrDoS-LDAP, and DrDoS-NTP, and the Exploitation based category contains DrDoS-UDP, SYN, and UDPLag.

B. Data Preprocessing

Data preprocessing is the initial stage of developing a machine learning model, involving the transformation of raw, unstructured data into a suitable format for modeling. As our research has been done on robust data, we have taken a batch size of 100000. As our research has been done on robust data, we have taken a batch size of 100000.

1) Batch Size: In machine learning, batch size is one of the primary hyperparameters that can make a great impact on the model training for data. To ensure the model performs at their best, batch size can be one of the most important measures. Also it can be seen how the increasing and decreasing of batch size can matter during the training of the model. [8]

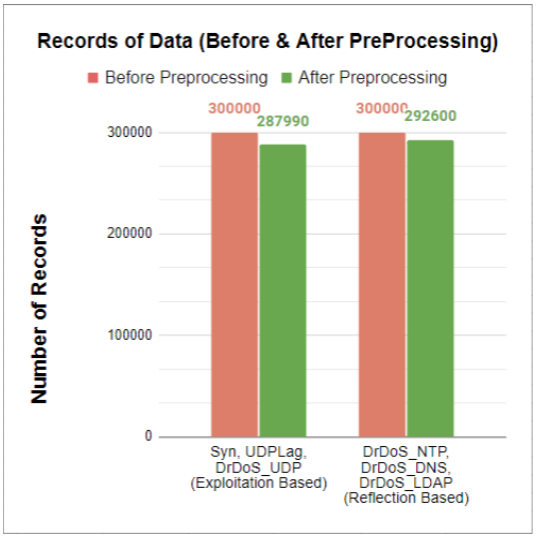

Fig. 2. Records of Data (Before and After PreProcessing)

After taking the batch size, the datasets were described separately where the actual data records were more than 3000000 on average. So the three datasets have been combined for each type. After merging them, the new dataset was formed shown in Fig. 2 where 3000000 data were taken for the preprocessing. A key part of this process is checking missing values and duplicate values in the sample. Most real-world datasets are prone to missing, inconsistent, or noisy data due to their diverse sources [9]. For example, missing or duplicate values can skew the overall statistics, leading to inaccurate insights [10]. In this study, a total 24020 number of null values were found and the null values were removed as there was a huge amount of data. Then the second most important step is Encoding which makes the data readable by the machine. As there are various types of data, such as numerical data, objective data, categorical data, some ML models can not take all the data types. Generally if there are categorical or objective type data which represent qualitative attributes in the dataset, ML models like Random Forest, Decision Tree can not recognize this type of data. So the data must need to be encoded for the model implementation. Label Encoding was performed in this work. Label encoding transforms the categorical data into meaningful numerical data.

2) Feature Selection: Feature selection is an essential pre-processing step in machine learning which is performed to find the lowest amount of features or attributes that boosts the performance of the models. In addition to improving accuracy, feature selection helps to build models more simply and to work quickly by using the minimum features [11]. In this study, Extra Tree Classifier have been performed for feature selection which is a type of Ensemble learning technique. Also, it is similar to a Random Forest Classifier. Before applying the feature selection method, there were 80 features in the datasets for both type. For this experimentation, the target variables were "Label", "SimilarHTTP", "Unnamed" etc. By applying this method, the best 20 features have been identified for further experimentation. The selected features were 'Timestamp', 'ACK Flag Count', 'Fwd Total', 'Fwd IAT Total', 'Flow Duration', 'Fwd Packet Length Min', 'Protocol', 'Flow ID', 'Source Port', 'Source IP' The data types were found int64, float64 and object.

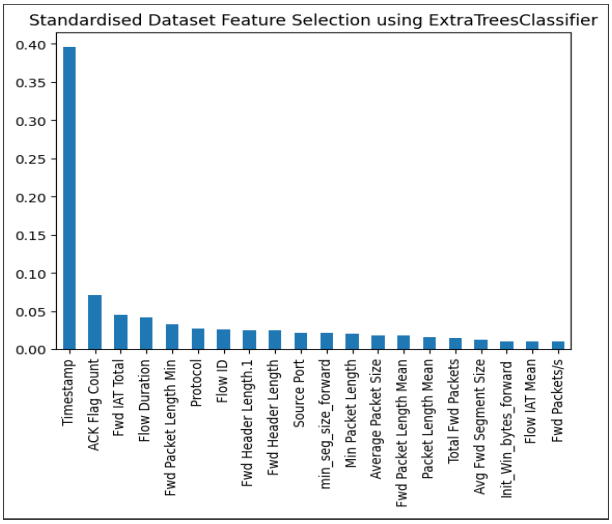

Fig. 3. Best 20 Features (Exploitation Type)

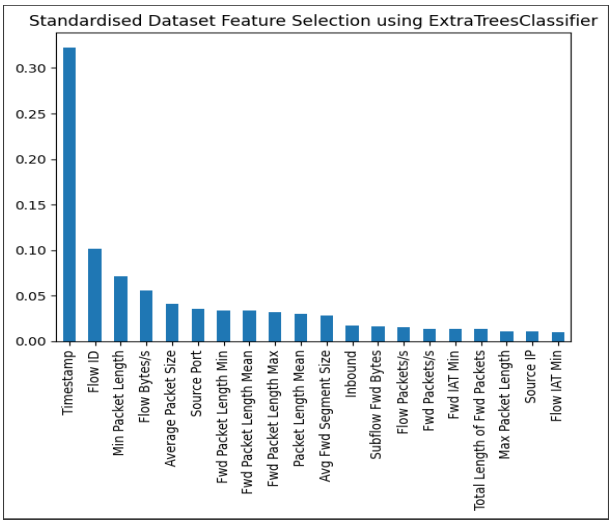

Fig. 4. Best 20 Features (Reflection Type)

The top most five features from the selected features from both type have been analyzed which are Timestamp, Fwd Packet Length Min, Ack Flag Count, Flow ID and Flow Duration.

Timestamp: A timestamp is a data field used to record the exact date and time, time intervals of an event occurrence. It is important to track the duration of specific network activities and synchronize events. Timestamps help to correlate and analyze traffic patterns to detect attacks. For example, an IDS can use this data field to identify repetitive traffic associated

with DDoS attacks.

Ack Flag Count: The Ack Flag Count refers to the total number of packets in a network flow where the Acknowledgement (ACK Flag) is set in the header of a TCP packet to indicate that the sender has successfully received data from the other side of the connection. In DDoS, malicious attackers can exploit the ACK flag where they send a large number of fake ACK packets to overwhelm a target server.

Flow ID: Flow Id denotes a unique data field which is basically a flow of packets that share the same characteristics between a source and a destination in a network. In network security, Flow ID helps in tracking and analyzing network traffic patterns. Also each flow can indicate a distinct communication event.

Fwd packet Length Min: Forward Packet Length Min refers to the smallest packets sent from the sender to the server in a network flow. It can help analyze normal traffic patterns as well as attack detection like DDoS or port scanning.

Flow Duration: Flow Duration refers to the time length that a specific network session lasts. It is identified from the transmission of the first packet and the last packet of a network flow. It is also important for analyzing traffic patterns, such as longer durations can sometimes be the cause of signal security issues like DDos.

C. XAI Feature Implementation

In this study, an XAI technique has been performed. Explainable AI (XAI) refers to the methods in ML that make the training and testing process of the AI models transparent and understandable to users. Basic ML models like Ensemble models (Random Forest, XGBoost) are often difficult to implement. XAI techniques aim to address this issue and provide insights on how and why a model makes predictions [12]. In this research, the SHAP library has been used for the random Forest classifiers showing which features are most influential in the models' prediction shown in Fig. 5 and Fig. 6. SHAP (SHapely Additive exPlanation) is a common XAI method that provides interpretable insights on ML model predictions. SHAP assigns a value to each feature based on Shapley values from cooperative game theory [13]. Most importantly, SHAP can be used for all type of ML models, such as linear model, linear model and Deep Learning model neural network.

1) Type Exploitation: The key explainable features obtained from SHAP include Timestamp, Fwd-Packet-Length-Mean, Ack-Flag-Count, Init-Win-Bytes-Forward, Source-Port and the others. It can be shown that some of the features differ from the initially selected features.

2) Type Reflection: Type Reflection: The identified features for type Reflection are Timestamp, Flow-ID, Min-Packet-Length, Fwd-Packet-Length-Min, Source IP and the other features. Here, the top 20 features remain consistent with the features that have been selected without the SHAP method.

D. Machine Learning Models

1) Random Forest: Random Forest is an Ensemble learning classifier that functions by building a multitude of decision

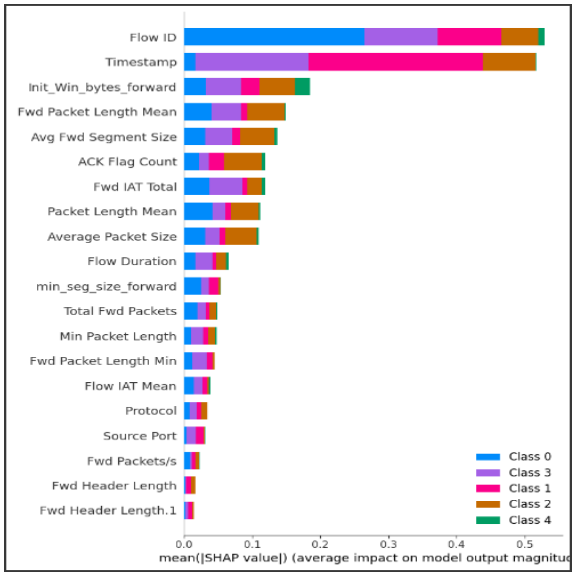

Fig. 5. Explainable AI features after K-fold cross validation.

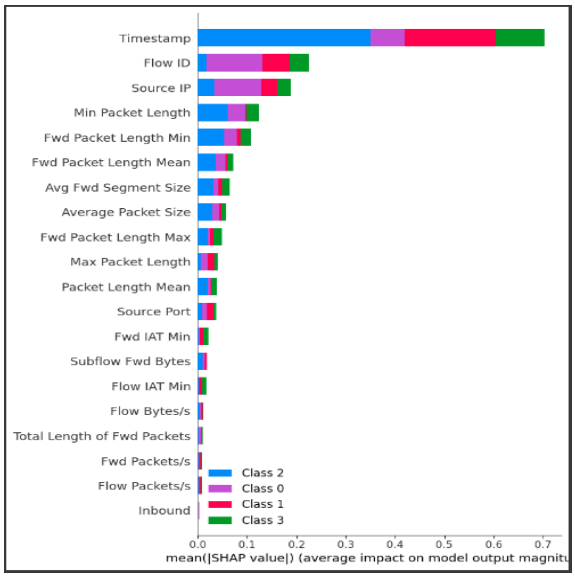

Fig. 6. Explainable AI features after K-fold cross validation.

trees at training time. This model can handle a large number of data, complex relationships between features and deal with missing data. This classifier performs well with high-dimensional data by making it suitable for implementation [14]. Also it tends to reduce the risk of overfitting. Random forests improve bagging because they de-correlate trees by introducing a split of features into random subsets.

$$\text{Gini} = 1 - \sum_{i=1}^{C} (p_i)^2 \tag{1}$$

$$\text{Entropy} = -\sum_{i=1}^{C} p_i \log_2(p_i) \tag{2}$$

2) Decision Tree: Decision Tree is another Ensemble based classifier in machine learning which is easy to interpret and suitable for both classification and regression tasks. Also it can handle non-linear data. Besides, this model does not require normalization as it can handle both numerical and categorical data.

$$\mathrm{IG}(D, A) = H(D) - \sum_{v \in \mathrm{Values}(A)} \frac{|D_v|}{|D|} H(D_v) \tag{3}$$

3) Naive Bayes: Naive Bayes is a linear classifier with a simple yet powerful functionalities in machine learning which is basically a form of Bayes' theorem. As it is a probabilistic classifier, it is well-fitted for high-dimensional data. Also it can handle both continuous and numerical data. It performs better than other models if the feature selection method can be performed accurately [15].

4) XGBoost: XGBoost classifier is an implementation of the eXtreme Gradient boosting algorithm which is an ensemble method. This model is known for its high performance, scalability and efficiency on structured data. Also it can perform at a fast speed and handle large datasets. Most importantly, this model provides built-in cross validation to help fine-tune models and also prevents overfitting during training [16]. Using the formula, the probability of the characteristics is calculated.

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^{t} \Omega(f_k) \tag{4}$$

IV. Experimental Result and Discussion

Google Collaboratore was used in this research for experimentation. The implementation was started with some large datasets split into two types, each containing three subsets. After combining the datasets and applying data preprocessing, the models including Random Forest, Naive Bayes, Decision Tree and XGBoost were trained and tested before and after feature selection. The ratio of train and test split was 60:40. Then the performance metrics like accuracy, precision, recall and f1-score have been evaluated. Finally, K-fold cross validation was performed to check the overfitting.

A. Result Analysis

The accuracy scores and training times for all the models are shown in TABLE I. It can be seen that Random Forest has achieved the highest accuracy for both types whereas Naive Bayes has the lowest accuracy. The highest accuracy for Exploitation based attacks is 99.89%(RF), and for Reflection based attacks is 99.74%(RF). Also, DT and XGBoost models have performed well according to the performance metrics. On the other hand, the lowest accuracy is 48.53% that has come from Naive Bayes model for type Exploitation. Another key outcome of this study is the reduced runtime compared to other works. As six datasets were split into two files, running these datasets in parallel resulted in improving training times. It was also observed that running the datasets individually, one after the other, would have significantly increased the overall runtime.

TABLE I
Comparison of Accuracy and Training Time among the four models.

Attack Types	Models	Accuracy (Batch size 100000)	Accuracy (Batch size 200000)	Training Time
Exploitation Based	Random Forest	0.9964	0.9989	27s
Exploitation Based	Naive Bayes	4853	0.5843	0.57s
Reflection Based	Random Forest	0.99748	0.9999	24s
Reflection Based	Decision Tree	0.9919	0.9974	54s
Reflection Based	XGBoost	0.9907	0.9964	34s

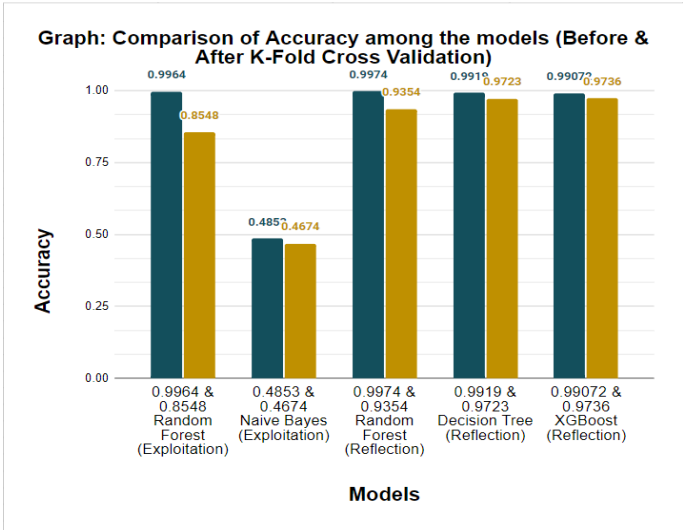

Fig. 7. Comparison of Accuracy among the models (Before and after performing Cross Validation)

Then, overfitting was checked by applying a K-fold cross validation process that is shown in Fig. 7. For Exploitation based and Reflection based attacks, the average cross validation scores are 85.48%(RF) and 93.54%(RF). Here it can be seen that the other models except RF and NB, performed well with the cross validation process. The average cv-scores of DT and XGBoost are 97.23% and 97.36% respectively.

TABLE II
Cross Validation Scores and Number of CV Scores used in Average (RF)

Average CV Scores	Number of CV Scores used in Average
77.07%	5
85.49 %	10
91.58%	14
93.59%	17
94.1%	21

From Figure 7, the difference of accuracy before and after applying cross validation for RF model can be noticed. So another batch size(200000) was taken for further experimentation. Also multiple folding values for splits were taken and the process was performed again. According to the outcome from TABLE II, if k is decreased , the accuracy become less whereas k is increased the accuracy and other metrics get higher for the models.

TABLE III
COMBINED ANALYSIS OF THE MODEL RANDOM FOREST FOR TYPE EXPLOITATION.

Exploitation Based	SYN	UDP_Lag	DrDOS_UDP	Benign	Accuracy
Precision	1	1	0.99	1	
Recall	1	1	1	1	0.99
F1-Score	1	1	0.99	1	

The combined categorization report for the Random Forest model for both type of attacks is shown in TABLE III and TABLE IV. It includes accuracy, support values, precision, recall, the f1-score across all classes and the accuracy. This analysis produces the highest scores of accuracy comparing to the other models. Lastly, some previous research using the same datasets was analyzed and their detection accuracy scores were compared to ours, as shown in TABLE V. This work achieved higher accuracy, with 99.89% for exploitation-based attacks and 99.748% for reflection-based attacks, surpassing the results of earlier studies.

TABLE IV
COMBINED ANALYSIS OF THE MODEL RANDOM FOREST FOR TYPE REFLECTION.

Reflection Based	DrDOS_DNS	DrDOS_LDAP	DrDOS_NTP	Benign	Accuracy
Precision	1	1	1	1	
Recall	1	0.99	0.99	1	0.99
F1-Score	1	1	0.99	1	

TABLE V
COMPARISON WITH OTHER STUDY ON SAME DATASET.

Authors	Models	Accuracy
Aktar et al. [4]	Deep Learning	93.41
Kumar et al. [5]	LSTM	98.00
Sayed et al. [6]	CNN, LSTM, GRU	89.4
Shieh et al. [17]	BI-LSTM, GMM	up to 94
Proposed Method	Random Forest	99.89, 99.748

V. CONCLUSION

DDoS attacks are not only disruptive nowadays, but also costly. To detect and prevent this threat, researchers are increasingly turning to advanced cyber security measures like machine learning and artificial intelligence. This research focuses on choosing the datasets and making it more suitable for preprocessing, analysing the features of the datasets and building some ML based models to detect the attacks accurately in a shorter time. The study shows that feature selection process is one of the most crucial part for this research, also models' performance evaluation varies in different state such as feature selection process, XAI technique and cross validation process. By combining real-time monitoring, automated defense mechanisms and multi-threading approaches, we can further defend against the evolving threat of DDoS attacks.

REFERENCES

[1] K. B. Dasari and N. Devarakonda, "Tcp/udp-based exploitation ddos attacks detection using ai classification algorithms with common uncorrelated feature subset selected by pearson, spearman and kendall correlation methods," *Revue d'Intelligence Artificielle*, vol. 36, no. 1, pp. 61–71, 2022.

[2] P. S. Saini, S. Behal, and S. Bhatia, "Detection of ddos attacks using machine learning algorithms," in *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 16–21, IEEE, 2020.

[3] M. Al-Shareeda, S. Manickam, and M. Ali, "Ddos attacks detection using machine learning and deep learning techniques: analysis and comparison," *Bulletin of Electrical Engineering and Informatics*, vol. 12, pp. 930–939, 04 2023.

[4] S. Aktar and A. Y. Nur, "Towards ddos attack detection using deep learning approach," *Computers & Security*, vol. 129, p. 103251, 2023.

[5] D. Kumar, R. Pateriya, R. K. Gupta, V. Dehalwar, and A. Sharma, "Ddos detection using deep learning," *Procedia Computer Science*, vol. 218, pp. 2420–2429, 2023.

[6] M. I. Sayed, I. M. Sayem, S. Saha, and A. Haque, "A multi-classifier for ddos attacks using stacking ensemble deep neural network," in *2022 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 1125–1130, IEEE, 2022.

[7] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," *ICISSp*, vol. 1, pp. 108–116, 2018.

[8] Devansh, "How does Batch Size impact your model learning — medium.com." https://medium.com/geekculture/how-does-batch-size-impact-your-model-learning-2dd34d9fb1fa. [Accessed 17-09-2024].

[9] "Data Preprocessing in Machine Learning: A Beginner's Guide — simplilearn.com." https://www.simplilearn.com/data-preprocessing-in-machine-learning-articlefaqs. [Accessed 29-08-2024].

[10] "Data Preprocessing in Machine learning - Javatpoint — javatpoint.com." https://www.javatpoint.com/data-preprocessing-machine-learning. [Accessed 29-08-2024].

[11] Y. Saeys, T. Abeel, and Y. Van de Peer, "Robust feature selection using ensemble feature selection techniques," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2008, Antwerp, Belgium, September 15-19, 2008, Proceedings, Part II 19*, pp. 313–325, Springer, 2008.

[12] "What is Explainable AI (XAI)? — IBM — ibm.com." https://www.ibm.com/topics/explainable-ai: :text=Explainable [Accessed 29-08-2024].

[13] F. Dallanoce, "Explainable AI: A Comprehensive Review of the Main Methods — dallanoce.fd." https://medium.com/@dallanoce.fd/explainable-ai-a-complete-summary-of-the-main-methods-a28f9ab132f7. [Accessed 29-08-2024].

[14] "Random Forest Algorithm — simplilearn.com." https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm. [Accessed 08-09-2024].

[15] "Naive Bayes Classifier in Machine Learning - Javatpoint — javatpoint.com." https://www.javatpoint.com/machine-learning-naive-bayes-classifier. [Accessed 08-09-2024].

[16] "What is XGBoost? An Introduction to XGBoost Algorithm in Machine Learning — Simplilearn — simplilearn.com." https://www.simplilearn.com/what-is-xgboost-algorithm-in-machine-learning-article. [Accessed 08-09-2024].

[17] T.-T. Nguyen, C.-S. Shieh, C.-H. Chen, and D. Miu, "Detection of unknown ddos attacks with deep learning and gaussian mixture model," in *2021 4th International Conference on Information and Computer Technologies (ICICT)*, pp. 27–32, IEEE, 2021.