Mesopotamian journal of Cybersecurity
Vol.4,No.3, **pp**. 278–290
DOI: https://doi.org/10.58496/MJCS/2024/027; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

Cybersecurity Defence Mechanism Against DDoS Attack with Explainability

Alaa Mohammed Mahmood [1,*], İsa Avcı [1]

[1]*Computer Engineering, Karabuk University, Karabuk, 78000, Turkey*.

ARTICLEINFO

Article history

Received 03 Oct 2024
Accepted 04 Dec 2024
Published 26 Dec 2024

Keywords

Application layer attack

DDoS detection

DDoS Mitigation

SHAP

ABSTRACT

Application-layer attacks (Layer 7 attacks), a form of distributed denial-of-service (DDoS) aimed at web servers, have become a significant concern in cybersecurity because of their ability to disrupt services by overwhelming server resources. This study focuses on addressing the challenges of detecting and mitigating the impact of such attacks, which are difficult to counter due to their sophisticated nature. The primary objective of this study is to develop an effective monitoring and defence model to detect, defend, and respond to these attacks efficiently. To achieve this, SHapley Additive exPlanations (SHAP) technology was used to understand the behaviour of the model and to increase the efficiency of the detection classifiers. The defence model is designed with three states: normal, observing, and suspicious. The observing mode, which represents the detection part, is triggered when the server load exceeds a predefined threshold. The detection system incorporates five machine learning (ML) algorithms: decision trees (DTs), support vector machines (SVMs), logistic regression (LR), naive Bayes (NB), and K-nearest neighbours (KNNs). A stacked classifier (SC) was then employed to combine these models to achieve optimal performance. The algorithms were evaluated in terms of accuracy (ACC), precision (PRC), recall (REC), F1 score (F1), and time (T). The SC demonstrates superior accuracy in distinguishing between legitimate traffic and malicious traffic. If the server continues to suffer from overload, the suspicious part of the defence model will be activated, and the mitigation algorithm will be called, which, in turn, bans users responsible for the attack and prevents illegitimate users from connecting to the server. The effects of the mitigation algorithm were noticeable in the server traffic rate, transmission rate, memory utilization, and CPU utilization, confirming its ability to defend against application-layer attacks.

1. INTRODUCTION

Distributed denial of service (DDoS) attacks has become a disturbing, widespread threat in today's cyber environment. These attacks aim to overwhelm and disable the targeted systems by flooding a network, website, or online service with excessive traffic or malicious requests, rendering them inaccessible to legitimate users [1]. The motivations behind these attacks can vary depending on the specific circumstances. In some cases, attackers use DDoS as a means of extortion, threatening organizations with disruption unless a ransom is paid; others may have ideological or political agendas to disrupt services or make a statement [2, 3]. In other cases, DDoS attacks may be used to damage the competition between companies but could also be used as a mask for other malicious activities, such as data breaches and network intrusions [2, 3]. Regardless of this motive, DDoS attacks disrupt normal operations, causing significant downtime, financial losses, and harm to organizations' reputation, resulting in long-lasting negative effects on the affected organizations and companies [4]. The selected attacks include the 2016 Dyn attack, the 2018 GitHub attack, and the 2016 KrebsOnSecurity attack, which disrupted services and websites in different ways [14–16].

Botnets, which are essential for executing DDoS attacks, can grow through several techniques, such as malware infections, phishing attacks, and exploiting weaknesses in networked devices. Once a botnet is established, attackers can launch DDoS attacks via compromised devices to flood targeted servers with continuous requests or data packets [9]. This overwhelming influx of traffic will deplete the critical resources of the system, such as bandwidth, memory, and processing power, leading to slowing system performance or even a complete service outage [10].

DDoS attacks can target various layers of the network stack by exploiting specific vulnerabilities. The nature of these attacks and the methods employed depend on the layer being targeted (Fig. 1) [11]. Common types of DDoS attacks include

*Corresponding author. Email: alaamahmood526@gmail.com

volumetric attacks, which exhaust the target's available bandwidth by overwhelming it with a massive volume of traffic. In volumetric attacks, attackers often leverage amplification techniques or botnets to generate enormous amounts of data, crippling network infrastructures. Other common DDoS attacks include TCP/IP Protocol attacks, which exploit vulnerabilities within the TCP/IP protocol stack and target weaknesses in protocols such as TCP, UDP, or ICMP. Common examples include SYN flood attacks, UDP flood attacks, and Ping flood attacks. Conversely, application layer attacks focus on exploiting vulnerabilities at the application layer of the target system. They deplete server resources by targeting specific application functionalities, such as through HTTP floods or DNS query floods [12- 13].

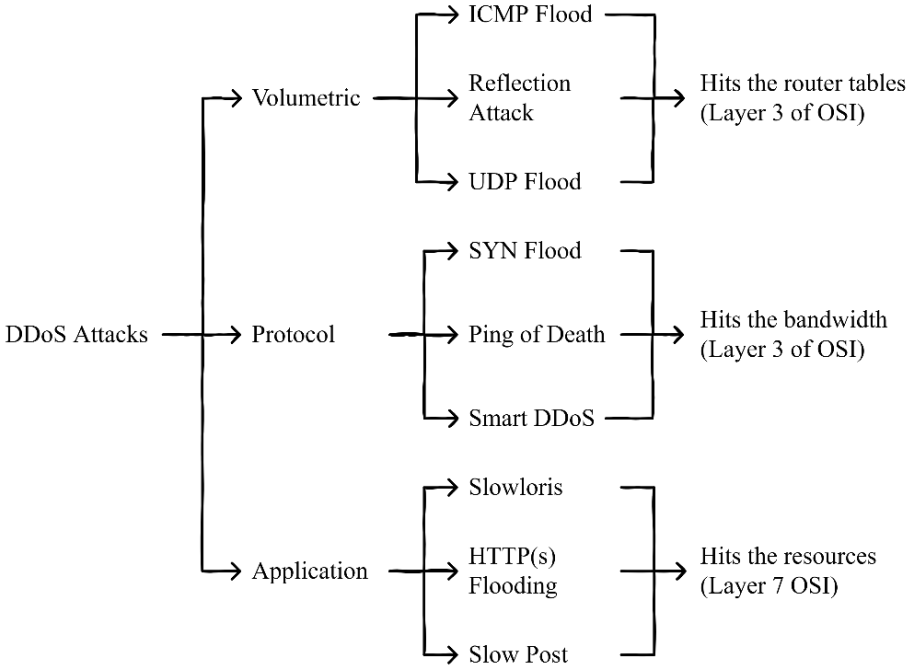

Fig. 1. Common types of DDoS attacks

To address the challenges posed by DDoS attacks, this study uses SHAP technology to interpret machine learning model outputs. SHAP is based on Shapley values, a game theory concept that assigns responsibility for a model's prediction to individual features or their specific values [21]. The characteristic feature of SHAP is its model-agnostic nature, allowing it to be applied to any machine learning model. SHAP provides consistent interpretable explanations and can effectively address complex model behaviours, such as feature interactions. This makes SHAP an invaluable tool for analysing sophisticated models [22]. SHAP offers human-readable explanations for predictions made by machine learning algorithms. It assigns a value to each input feature, indicating its contribution to the final prediction. This helps teams understand the decision-making process of the model and identify the most significant factors influencing its output [31].

The main motivation for this study is that many institutions, companies, and state-affiliated websites continue to suffer from DDoS attacks. Despite advancements in artificial intelligence, there has yet to be a comprehensive classification of algorithms based on multiple criteria. For example, it cannot be assumed that an algorithm with high accuracy, such as Algorithm X, can be fully trusted, as its performance in this context relies on various factors, including accuracy, processing time, and resource efficiency. This study addresses the critical challenge of application-layer (Layer 7) DDoS attacks, which are particularly dangerous because their ability to mimic legitimate traffic makes it extremely difficult to differentiate between legitimate and malicious users. The goal is to detect and mitigate these attacks while ensuring that the server remains operational without exhausting its resources, such as memory and processing power.

This study fills the gaps in existing research by focusing on classifier selection and multicriteria evaluation for DDoS detection and aims to improve detection accuracy and identify the most suitable classifiers that consider multiple performance metrics. The goal is to develop an effective method to detect, mitigate, and potentially eliminate the impact of DDoS attacks on web servers. The significance of this study lies in the importance for companies to adopt robust mechanisms such as traffic monitoring, anomaly detection, and traffic filtering while also familiarizing themselves with effective mitigation techniques to counteract the impact of DDoS attacks. Collaboration and information exchange among companies, institutions, and internet service providers (ISPs) are equally important. These efforts, combined with a strong commitment to combat attackers, can significantly enhance the ability to quickly identify, mitigate, and potentially eliminate DDoS attacks [5-23]. Understanding the nature of DDoS attacks helps organizations improve their defences and

protect against serious consequences [6-32]. By utilizing specialized DDoS mitigation systems, monitoring traffic, and implementing postattack strategies, organizations can reduce the impact of such attacks. Partnering with ISPs or specialized services can further increase security [7-19]. This study provides valuable insights to help organizations, especially those involved in state security, improve their defenses and better protect against cyber threats.

The key contributions of this research include improving detection accuracy by enhancing deep learning algorithms with ensemble learning techniques combined with SHAP, identifying the most suitable algorithm based on both accuracy and execution time among a range of options, and developing a real-time mitigation algorithm and deploying it on a local server for immediate application.

2. RELATED WORK

Cynthia et al. (2023) utilized SHAP to select features for DDoS attack detection. Using the CICIDS2017 dataset, researchers have demonstrated that the SHAP technique enhances model interpretability and efficiency. Additionally, they employed a conditional tabular GAN (CTGAN) to generate synthetic data, which facilitated the training of an improved classifier. Their model achieved high accuracy, with a random forest (RF) accuracy rate of 99%. However, the key limitation in their work was the need for real-time data to test the practical performance of SHAP across different conditions. Cynthia et al. recommended further research to improve synthetic data and detection methods for real-time environments [36].

Akinwale et al. (2024) presented "A Regenerative Model for Mitigating Attacks on HTTP Servers for Mobile Wireless Networks," which focuses on the strength of the HTTP protocol. The CICIDS2017 dataset and techniques such as SMOTE, random sampling, random dropout, and principal component analysis were used. Akinwale et al. (HReg) demonstrated robust defence against SQL injection and DoS attacks, enhancing mobile network security. However, researchers have highlighted the need for real-world data to evaluate model performance. They also recommend the use of firewalls and continuous monitoring to ensure long-term reliability in network environments [29].

Dogra and Taqdir (2024), in their work "Enhancing Detection of Distributed Denial of Service Attacks and Network Elasticity through Packet Processing and Frequency Range Optimization," employed random forest algorithms to analyse network traffic and optimize frequency ranges. Their group-based approach significantly reduces packet rates, improving network elasticity and resistance to DDoS attacks. However, the effectiveness of their model decreased with more complex attack patterns, which remain underexplored. The authors recommend further testing in diverse network settings to increase adaptability and reliability [30].

Tedyyana et al. (2024) developed "Automated Learning for Network defence: Real-Time Detection of DDoS Attacks with Telegram Notifications." which achieved 99.77% accuracy and an F1 score of 98.70% when DT, SVM, and neural networks were trained on the CICIDS2018 dataset. The integration of a Telegram-based notification system for real-time alerts enhances its practical application. However, reliance on Telegram limits integration with other notification protocols. The study recommends retraining the model with new attack data to adapt to evolving environments [31].

Bindu et al. (2024), in their study "Detection of DDoS Attacks in SDN Networks Using Machine Learning," utilized machine learning algorithms such as random forests, k-nearest neighbours, DT and LR to analyse network traffic. The authors demonstrated that combining software-defined networking (SDN) with machine learning offers an effective method for detecting and mitigating DDoS attacks. While the study highlights the importance of cooperative cybersecurity frameworks, they lack real-time application, which may limit their utility during active attacks. The authors recommend further research into advanced machine learning techniques to enhance detection abilities in cooperative security networks [33].

Layeq et al. (2024) investigated the application of Edge-IIoT networks and SMOTE for training ensemble learning models. They utilized techniques such as hard voting, soft voting, and stacking to improve detection rates for DDoS attacks in Edge-IIoT environments. However, class imbalance may affect model accuracy in real-world environments. Layeq et al. recommend addressing class imbalance issues and exploring broader IoT security challenges in future work [34].

Ahmed et al. (2019) and Osid et al. (2018) wrote comprehensive reviews on DDoS detection and mitigation techniques. Ahmed et al. compared statistical methods with machine learning-based approaches, whereas Osid et al. explored both traditional and advanced techniques, including data mining and anomaly detection. These reviews highlight the evolution of defence strategies against DDoS attacks, identifying strengths and weaknesses in various methodologies. Earlier

contributions by Antonakakis et al. (2011) and Rajab et al. (2006) provided foundational insights into signature extraction, attack classification, and trends that continue to influence modern cybersecurity research [24-28].

3. METHODOLOGY

The defence system modules are divided into three parts: natural, observing, and suspicious (Fig. 2). The observation mode depends on the comparison between the load and the threshold; if the load exceeds the predefined threshold, the ML algorithm runs, checks the traffic and detects if there is an attack. Once the system detects an attack, it transitions to Suspicious Mode after verifying that it is a genuine attack and not merely normal user behaviour (Fig. 2). In Suspicious Mode, if the load still exceeds the threshold, each user must pass a CAPTCHA test to join the server, thereby initiating the mitigation algorithm. This algorithm blocks every IP address that sends excessive traffic to the server and adds it to the blacklist. The blocked IP cannot be permanently blocked; instead, an initial period is given, which gradually increases if the IP continues to send high volumes of traffic. If the situation reverses, the IP will be removed from the blacklist and added to the whitelist.

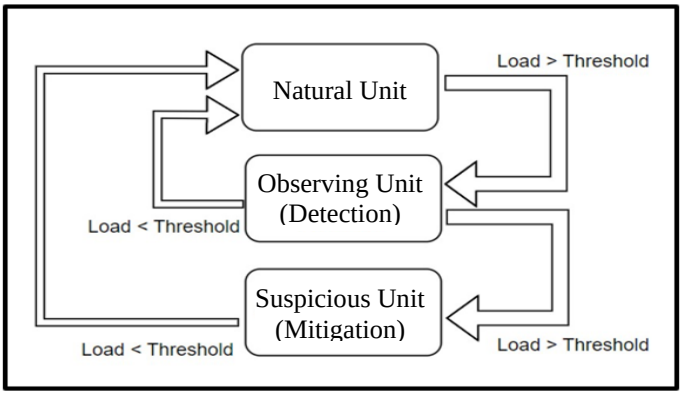

Fig. 2. Switching process between defense system units

3.1 Detection Stage

Fig. 3 shows the contents of the observation box in Fig. 2. To start the process, the CIC-DDoS2019 dataset is pre-processed by loading it and handling any missing values, either by filling them with a default value or removing them entirely. It is important to normalize the features to ensure that they are on the same scale, which helps algorithms work correctly. After preprocessing, the dataset is split into training and testing sets, with an 80–20 split. The test set should remain unseen during training for reliable evaluation.

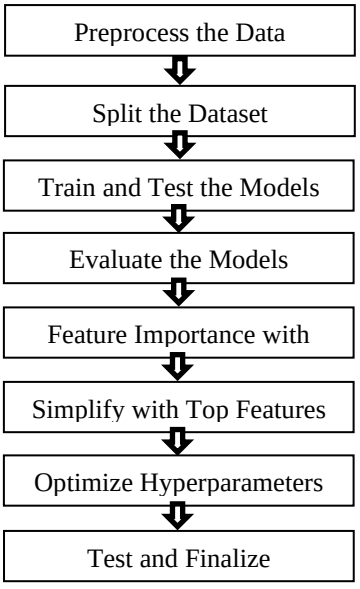

Fig. 3. Detection procedures

Next, the machine learning models DT, SVM, LR, NB, KNN, and stacked classifiers (a stacked classifier is an ensemble learning method that combines individual classifiers to enhance predictions [33]) are trained on the training set and then tested on the test set. Evaluation metrics such as accuracy (ACC), precision (PRC), recall (REC), F1 score, and implementation time are used to assess model performance. To improve the model's interpretability, SHAP is used to analyse features and identify the most important features. After the most important features are determined, only the top 10–15 features [32] are retained based on SHAP importance, and then the model is retrained with the reduced feature set. This will improve accuracy, as SHAP not only helps identify the most important features but also simplifies the model. To enhance performance, the model's hyperparameters are optimized by tuning parameters such as the number of estimators or the maximum depth of trees in a random forest model. This tuning helps identify the best combination of settings for maximum accuracy. Finally, the trained model is evaluated on the test set to obtain a high-performing and interpretable model for detecting DDoS attacks via the CIC-DDoS2019 dataset.

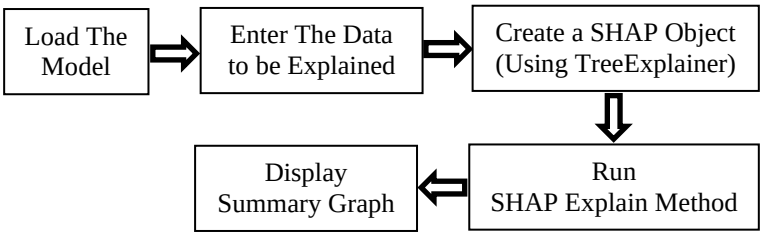

Fig. 4. SHAP implementation steps

Fig. 4 shows the SHAP implementation steps. The process begins by loading the trained model whose predictions need to be interpreted. Then, the data that need to be interpreted are entered. A SHAP object is then created via TreeExplainer. Interpretation is particularly suitable for tree-based models such as decision trees or random forests. Next, the SHAP explanation method is executed, and SHAP values representing the contribution of each feature to the prediction are calculated. Finally, a summary flowchart is shown, which provides an overview of the contributions of these features, which in turn helps users understand how each input feature affects the model's decision.

3.2 Mitigation

To trigger the system to enter the suspicious mode (mitigation), the load must exceed the threshold. As a precaution for this mode, new users who want to join the server must pass a CAPTCHA test, which helps prevent bot users from joining the server. A Slowloris application DDoS attack was applied to a Unix server connected to users with 2 GB of RAM and 1 GB of CPU. The traffic, transmission rate, CPU utilization, and memory utilization were all monitored and recorded. The attack was applied to two forces: the first was somewhat light, and the second was stronger than the first by increasing the number of packets/second. Once the attack is detected, the mitigation algorithm is activated to stop the attack. The mitigation process checks if the source IP address has already been recorded; if not, it will be included in the list, and the blocking period may be extended if a specific IP address resurfaces (see Fig. 5). Since some valid IPs may temporarily become entangled with zombie groups, the mitigation module does not permanently bar the IP address. When authorized users detect misused devices and perform security updates, the source IP address can once again become legitimate. Typically, the same network generates these zombie devices to launch a powerful DDoS attack. Attackers aim to gather a substantial collection of devices, often infecting a particular network to convert its devices into botnets targeting a specific objective.

The mitigation algorithm can obstruct IPs originating from the same network or subnet rather than blocking each source IP individually. Periodic updates and temporary maintenance of the blacklist, which expires within a predefined timeframe, protect legitimate users from future obstructions. The metrics used before and after the mitigation process to evaluate the impact of the attack included the CPU utilization, memory utilization, transmission rate, and traffic rate.

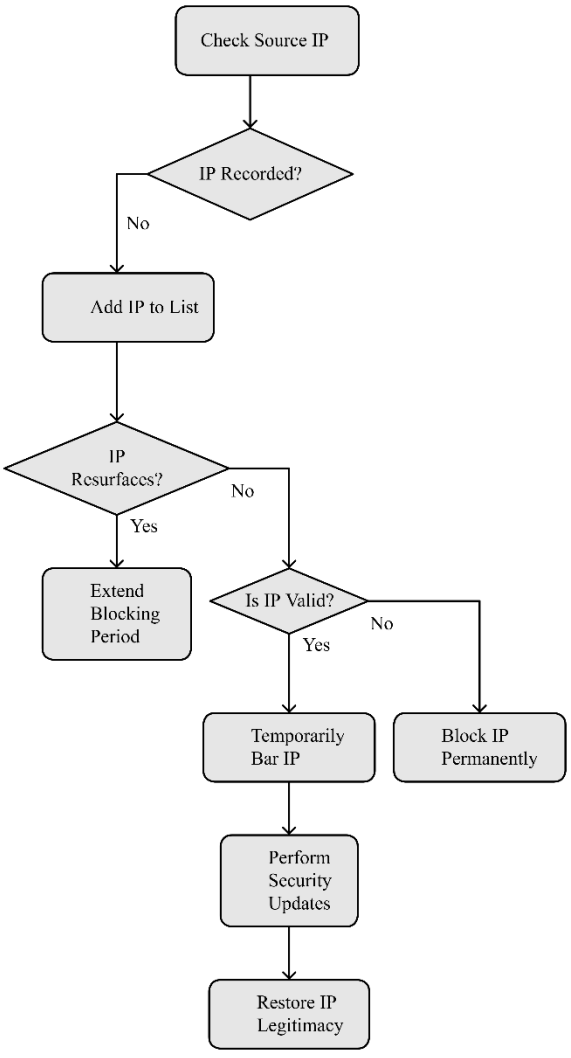

Fig. 5. Users IP management during the mitigation stage

4. RESULTS AND DISCUSSION

4.1 Detection Results

Five basic machine learning classifiers and a stacked classifier were built via the Python programming language. As depicted in Fig. 6, the support vector machine (SVM) classifier outperformed all the other base classifiers, achieving exceptional performance results, with an accuracy of 99.32%, precision of 99.33%, recall of 99.32%, and an F1 score of 99.32%. Logistic regression (LR) and k-nearest neighbour (KNN) followed closely. LR achieved an accuracy of 99.13%, precision of 99.14%, recall of 99.13%, and F1 score of 99.13%, whereas KNN achieved an accuracy of 97.82%, precision of 97.81%, recall of 97.82%, and F1 score of 97.84%. The decision tree (DT) classifier ranked fourth with good metrics, with an accuracy of 93.11%, precision of 93.12%, recall of 93.11%, and an F1 score of 93.11%. Conversely, the naive Bayes (NB) classifier showed more conservative performance, with lower values of 76.22% accuracy, 75.25% precision, 76.22% recall, and an F1 score of 80.02%. Consequently, the algorithms are ordered from best to worst performance as SVM > LR > KNN > DT > NB. In terms of execution time, the KNN algorithm was the fastest, whereas the DT algorithm took the longest.

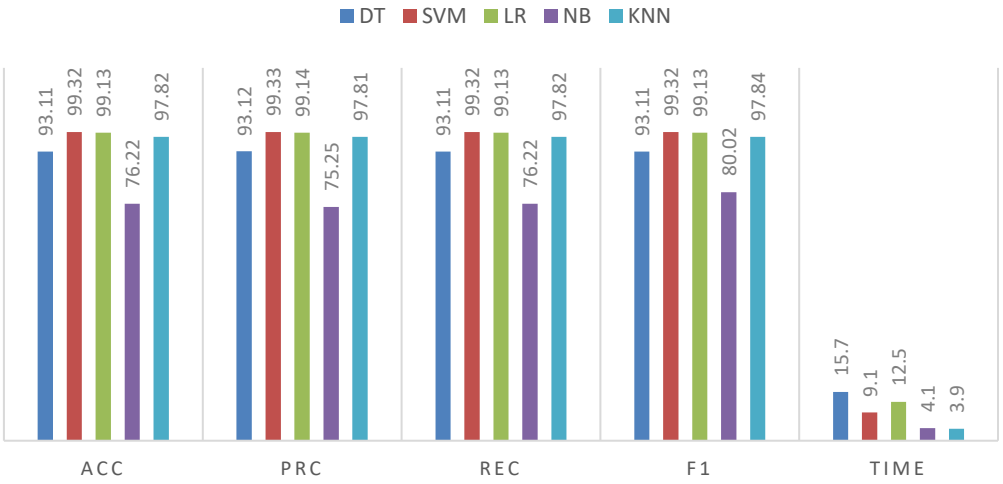

Fig. 6. ML algorithm results

The superiority of the linear SVM classifier over the other classifiers highlights the dataset's inherently linear nature. This conclusion arises from the linear SVM's ability to effectively establish clear boundaries among different categories via linear techniques. Therefore, classifiers based on the linear data separation technique are the best choice for this specific problem. In contrast, the naive Bayes (NB) classifier, which relies on a probabilistic approach, demonstrated the lowest performance with the given problem because it is incompatable with the dataset structure. This suggests the limitations of probability-based classifiers when applied to challenges of this nature.

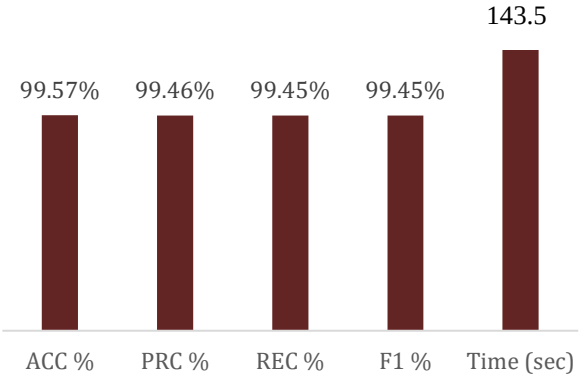

Fig. 7. SC (DT+SVM+LR+KNN) metrics

TABLE I. STACKED CLASSIFIER ACCURACY

Stacked Classifiers	Accuracy
DT+SVM+LR+KNN	99.57%
DT+SVM+LR	99.22%
KNN+NB	97.52%
LR+NB	98.78%
LR+KNN	98.73%
SVM + NB	99.17%
SVM+KNN	99.31%
SVM+LR	99.12%
DT+NB	92.86%
DT+KNN	98.34%
DT+LR	99.22%
DT+SVM	99.17%

The effectiveness of various combinations of machine learning classifiers on a given dataset is summarized in Table I. The classifiers include decision trees (DTs), support vector machines (SVMs), logistic regression (LR), k-nearest neighbors (KNNs), and naive Bayes (NB).

The highest obtained accuracy was 99.57%, which was achieved by stacking the DT, SVM, LR, and KNN classifiers; additional performance metrics for this stacked classifier are presented in Fig. 7. The DT + SVM + LR combination also shows good performance, with an accuracy of 99.22%, which is better than the result obtained from the DT + SVM combination without LR, with an accuracy of 99.17%. This suggests that the predictive ability of DT and SVM is further enhanced when they are added to the LR classifier. In fact, the LR appears consistently in high-performing stacks, indicating its strength across different combinations and its effectiveness in complementing other classifiers. Furthermore, combining the SVM with the KNN classifier achieved an accuracy of 99.41%, suggesting that these models can effectively handle complex data distributions when used together. On the other hand, the lowest accuracy was 92.86% for the combination (DT + NB), suggesting that NB may struggle to capture the dataset's complexities even when combined with DT, which typically models more intricate patterns. Overall, the results emphasize the importance of diversity in model selection for

stacking, as combinations that mix various types of models tend to produce better results than those with similar characteristics. This analysis confirms that utilizing a diverse set of classifiers in stacking can lead to significant enhancements in accuracy, highlighting the need for careful selection of model combinations to achieve optimal results. On the other hand, the implementation time for the combination of (DT + SVM + LR + KNN) was 143.5 sec, which is considered relatively long compared with the rest of the classifier implementation times (Fig. 6 and Fig. 7), and this conflicts with one of the interests of this work, in which the time factor is important for eliminating the attack as quickly as possible. In Table I, all the mentioned values except the value of "DT+SVM+LR+KNN" are less than the accuracy value of the SVM classifier (99.32%), so the work relies only on the value of the stacked classifier "DT+SVM+LR+KNN", as shown in Fig. 7, and there is no need for the remaining classifiers because their accuracy is less than the accuracy of the SVM; of course, their implementation time is larger than the SVM implementation time (9.1 sec) (see Fig. 6), which is due to the integration of more than one classifier. As the accuracy factor is important, the time factor is equally significant.

All the previous results were obtained before the SHAP method was applied. Next, the results after applying the SHAP method are presented and compared with those obtained prior to its use.

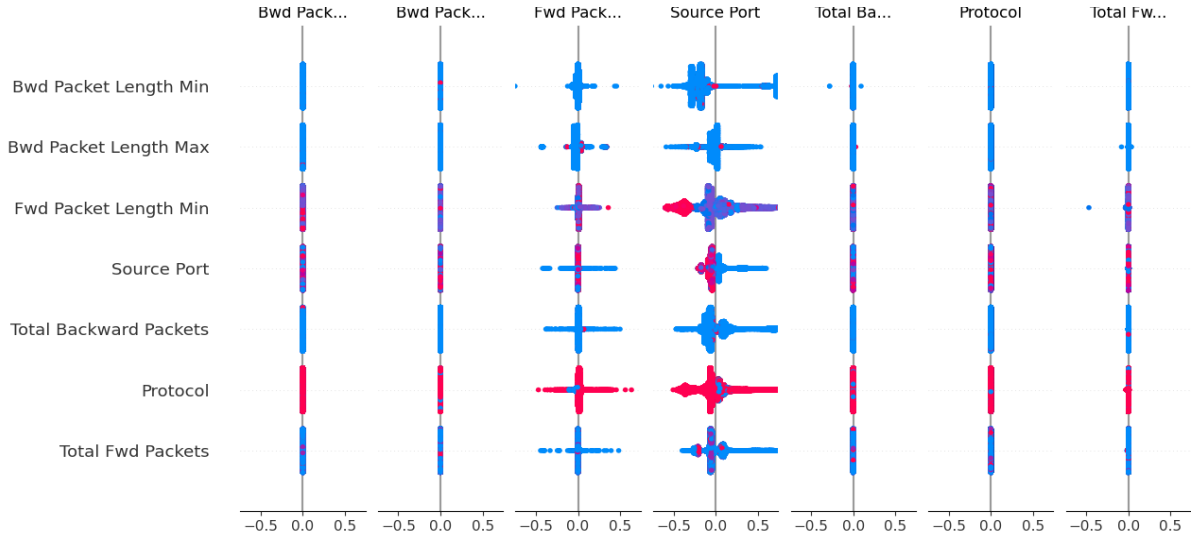

Fig. 8. SHAP values

Fig. 8 displays the SHAP graph values, illustrating how different features interact and affect predictions. The vertical axis represents packet lengths, port numbers, protocols, and other relevant features, whereas the horizontal axis indicates the SHAP interaction values, which reflect the effect of feature interactions on the model outputs. The dots represent individual data points, showing the direction of the effect, whether positive or negative, and their colors, ranging from blue to red, reflect the magnitude of the feature values. The 'Source Port' feature is represented by both red and blue dots. The red dots indicate higher values of the source port, which have a stronger impact (either positive or negative) on the predictions, whereas the blue dots represent lower values. The 'Source Port' feature exhibited the highest SHAP interaction values, making it the most impactful feature on the model's output. As a result, a greater focus on this feature, along with other similar features in the CIC-DDoS2019 dataset, will further improve the detection accuracy of the classifiers.

TABLE II. BEFORE SHAP

	ACC	PRC	REC	Fl	Time
DT	93.11%	93.12%	93.11%	93.11%	15.7
SVM	99.32%	99.33%	99.32%	99.32%	9.1
LR	99.13%	99.14%	99.13%	99.13%	12.5
NB	76.22%	75.25%	76.22%	80.02%	4.1
KNN	97.82%	97.81%	97.82%	97.84%	3.9

TABLE III. AFTER SHAP

	ACC	PRC	REC	Fl	Time
DT	95.12%	95.12%	95.11%	95.13%	8
SVM	99.81%	99.81%	99.82%	99.80%	4
LR	99.50%	99.54%	99.51%	99.51%	6.3
NB	81.13%	81.07%	80.99%	81.09%	2.2
KNN	98.10%	98.11%	98.09%	98.10%	2.5

From TABLE II and TABLE III, it is evident that after applying the SHAP method, the accuracy of all classifiers increased while their execution time decreased, aligning perfectly with our goals. This improvement is attributed to the ability of SHAP to isolate high-impact features and disregard those with minimal impact. Notably, the rankings of the classifiers remained consistent before and after the application of SHAP: SVM, LR, KNN, DT, and NB. The most significant improvement was observed in the NB algorithm, which experienced a substantial increase in accuracy. Overall, the SVM algorithm continues to stand out as the best because of its high accuracy and short execution time.

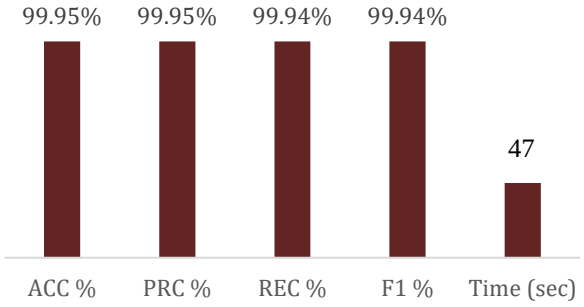

Fig. 9. SC (DT+SVM+LR+KNN) metrics after SHAP

TABLE IV. SC ACC AFTER SHAP

Stacking Classifiers	ACC
DT+SVM+LR+KNN	99.95%
KNN+SVM	99.81%
SVM+KNN	99.78%
DT+SVM	99.60%
LR+KNN	99.62%
DT+SVM	99.58%
SVM + NB	99.51%
SVM+NB	99.53%
SVM+LR	99.47%
DT+LR	98.71%
DT+KNN	98.51%
DT+ND	92.99%

As shown in TABLE IV and Fig. 9, the stacked classifier (DT+SVM+LR+KNN) achieved a significant improvement in accuracy, increasing from 99.57% to 99.95%, the highest accuracy achieved in this work. Additionally, its execution time decreased significantly from 143 seconds to 47 seconds. Although this reduction in execution time is impressive, it remains relatively greater than those of the individual classifiers. For the other classifier combinations listed in TABLE IV, the accuracy and execution time both improve; however, their accuracy remains lower than that of the single SVM classifier. Considering that both time and accuracy are critical for this study, the SVM classifier proves to be the most suitable choice, as it satisfies these two factors. However, if the time factor is less critical, the stacked classifier (DT+SVM+LR+KNN), which achieves the highest accuracy, would be the preferred choice.

Table V quickly compares the accuracy of different classification approaches for DDoS detection models across various classifiers and feature selection methods.

TABLE V. THE PERFORMANCE COMPARISON TABLE FOR THE PROPOSED MODEL WITH OTHER METHODS IN THE LITERATURE

Reference Papers	Feature Selection	Classifier	Accuracy
1. DDoS Attack Detection Using SHAP-Based Feature Reduction [32].	SHAP	CTGAN	99%
2. Enhancing the discovery of the deprivation attacks of the distributed service and the elasticity of the network by processing the group's packets and improving the frequency range [27].	Frequency Domain Analysis	RandomForest	95%
3. Automated Learning to Defending the Network: Automated Detection of Distribution attacks using Telegram notifications [28].	CICIDS2019	DT	99.77%.
4. Discovery of DDOS attacks in SDN networks using machine learning [29].	NetworkTraffic	RandomForest	99.91%
5. DDOS Attack Detection in Edge-IIOT Network Using Ensample Learning [30].	SMOTE	Ensemble	99.91%
6. This Work	SHAP	Stacking Classifer	99.95%

4.2 Mitigation Results

The implementation of the attack led to significant changes in memory usage, CPU performance, network traffic, and transaction rates. This is thoroughly illustrated in Figures 10 to 13. As a result of the attack, the system transitioned into a suspicious state, prompting the activation of the mitigation algorithm, as visualized in Figures 14 to 17. In Figures 10 to 17, the first attack (blue curves) was less powerful than the second attack (red curves).

4.2.1 Results Before Mitigation (During DDoS Attack)

These cases were recorded for one minute during the attack, and before invoking the mitigation algorithm, the CPU usage was 100%, as shown in Fig. 13, and almost 1 GB of memory (half of it) was utilized in Fig. 12. Additionally, there was a sharp and rapid rise in data transfer and traffic, as shown in Fig. 10 and Fig. 11.

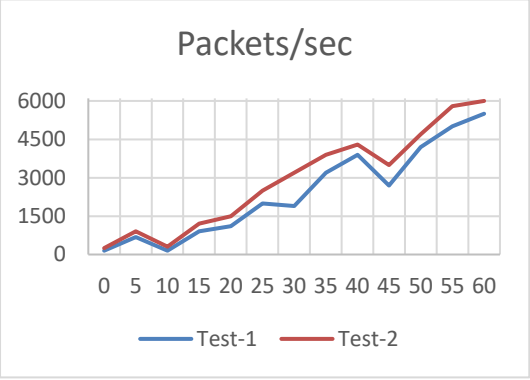

Fig. 10. Traffic rate

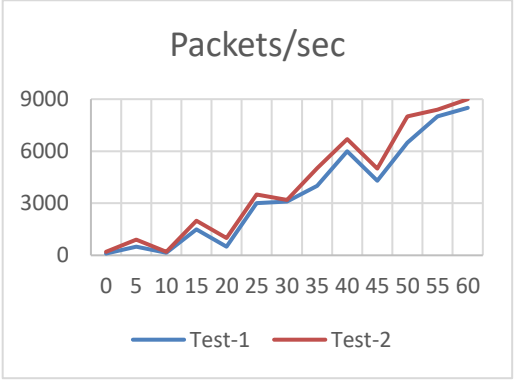

Fig. 11. Transmission rate

Notably, the traffic rate has exceeded 6000 packets/second, exceeding the server's limits or threshold, which has placed a heavy load on it and made it unavailable to users (Fig. 10). Additionally, the transmission rate increased rapidly, reaching 9000 packets per second, which is a high rate that exceeds the permitted limit for using server resources (Fig. 11).

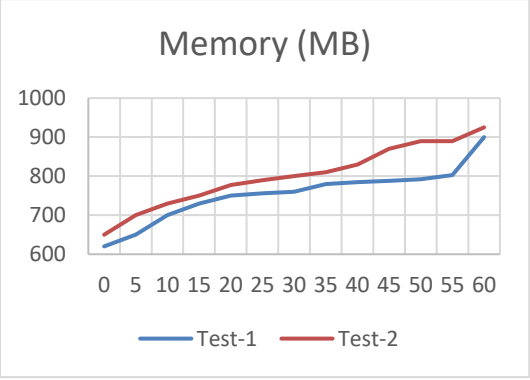

Fig. 12. Memory utilization

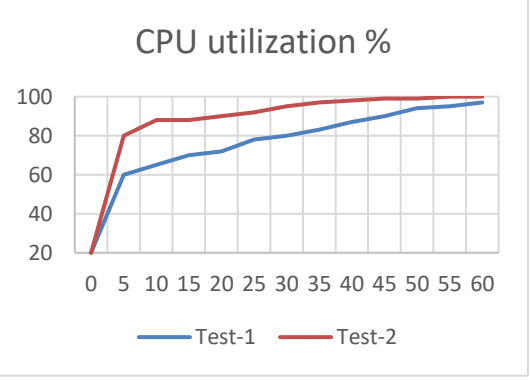

Fig. 13. CPU utilization

4.2.2 Results After Mitigation

After the mitigation algorithm is implemented, the processor, memory, and data transfer activities return to their normal states, as illustrated in Figures 14 to 17. The CPU usage normalized, reaching 20% of the total utilization, as shown in Fig. 17. Similarly, the memory usage stabilized at 200 MB (Fig. 16). Furthermore, the volume of data sent and received decreased to normal levels, as evident in Fig. 14 and Fig. 15.

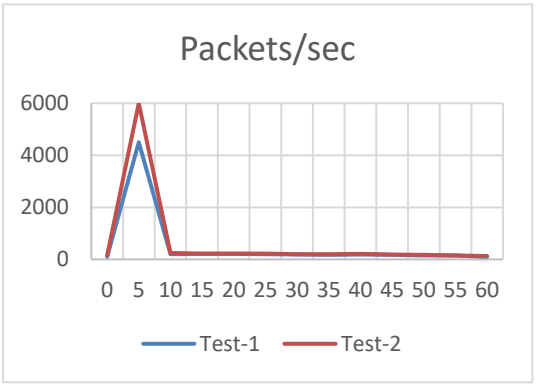

Fig. 14. Traffic rate

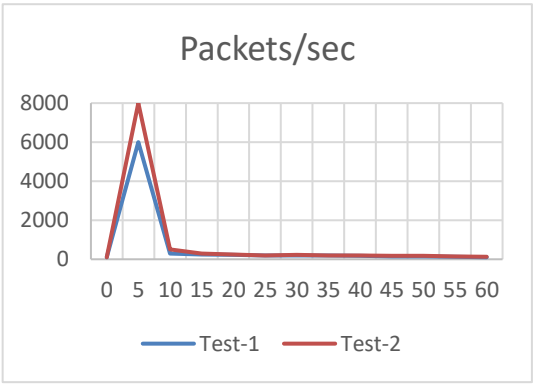

Fig. 15. Transmission rate

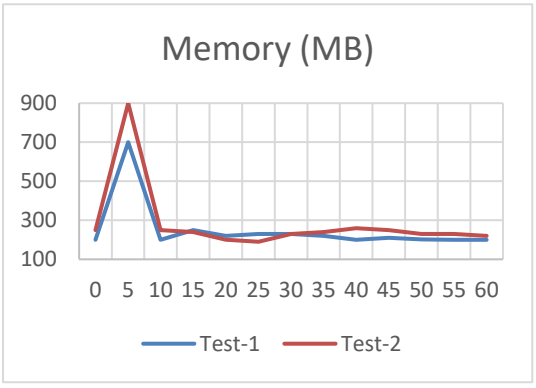

Fig. 16. Memory utilization

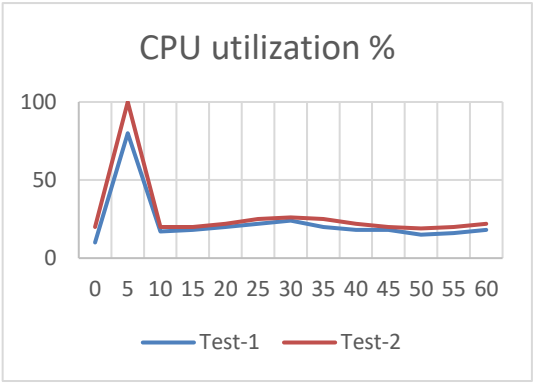

Fig. 17. CPU utilization

TABLE VI presents a summary of the metrics obtained before and after mitigation. The values in the table illustrate the effectiveness of this algorithm in controlling the attack. During the attack, 100% of the CPU was utilized, along with 90% of the total memory, indicating exhaustion of the server's resources during the attack. After mitigation, we observe that the CPU usage decreases to 18%, whereas the memory usage decreases to 20% of the total capacity.

TABLE VI. METRICS BEFORE AND AFTER MITIGATION

Metric	Before Mitigation	After Mitigation
Traffic Rate (Mbps)	5010	150
Transmission Rate	5500	100
CPU Utilization (%)	100	18
Memory Usage (MB)	900	200

5. CONCLUSIONS

The main idea presented in this study is to detect and mitigate DDoS attacks at Layer 7 via machine learning algorithms with the help of SHAP. This integration enhances the detection accuracy and reduces the implementation time of the classifiers. Unlike many existing studies that focus solely on accuracy, this work emphasizes balancing accuracy and execution time, which is crucial for detecting and mitigating attacks in real time or close to it. Additionally, the study introduces a mitigation algorithm that dynamically blocks and unblocks IP addresses based on traffic patterns. The practical implementation and testing on a local Unix server highlight the work, bridging the gap between theory and real-world application.

This study addresses several gaps in the literature, contributing to the field of DDoS defence. The stacked classifier, with the help of SHAP, demonstrated the highest accuracy in attack detection when multiple machine learning algorithms were combined, achieving an accuracy of 99.95%, albeit at the expense of execution time, which took 47 seconds. However, for tasks requiring a fast response in real time, the SVM algorithm, aided by SHAP, proved more suitable because of its strong balance between accuracy (99.81%) and execution time (4 s). This makes it ideal for applications where accuracy is critical, and execution time remains a crucial factor for real-time detection. Dividing the system into normal, observing, and suspicious states simplified the process and significantly contributed to monitoring and mitigating DDoS attacks.

Testing on real attacks confirmed that the mitigation algorithm successfully prevented malicious users from connecting to the server or overloading it, with the CPU and RAM restored to their normal states prior to the attack. The SHAP technique greatly enhances the understanding of model outputs and identifies important features in the CIC-DDoS2019 dataset, and its use is recommended for improved and faster learning.

One limitation of this study is its reliance on the CIC-DDoS2019 dataset, which may limit adaptability, particularly when faced with new attack patterns or advanced threats. Another consideration is that while the stacked classifier achieves high accuracy, its execution time is longer than that of other classifiers, potentially limiting its practical application in high-traffic, real-time situations where fast response is essential. The mitigation algorithm, which blocks IP addresses based on a predetermined threshold, may also require continuous adjustments to respond to changing traffic and server loads, reducing its effectiveness in dynamic environments. Finally, this study focused on Layer 7 distributed denial-of-service attacks. Generalizing the results to include other forms of distributed denial-of-service attacks would broaden the scope of this research and should be considered in future studies.

Conflicts of interest

The authors declare that they have no conflicts of interest.

Funding

No funding was received.

Acknowledgement

I want to thank everyone who helped with this work.

References

[1] S. Wani, M. Imthiyas, H. Almohamedh, K. M. Alhamed, S. Almotairi, and Y. Gulzar, "Distributed denial of service (DDoS) mitigation using blockchain—A comprehensive insight," Symmetry, vol. 13, no. 2, p. 227, 2021.

[2] M. Roopak, G. Y. Tian, and J. Chambers, "Multi-objective-based feature selection for DDoS attack detection in IoT networks," IET Networks, vol. 9, no. 3, pp. 120–127, 2020.

[3] Mishra, B. B. Gupta, and R. C. Joshi, "A comparative study of distributed denial of service attacks, intrusion tolerance and mitigation techniques," in Proc. European Intelligence and Security Informatics Conf., pp. 286–289, Sep. 2011.

[4] Abhishta, W. van Heeswijk, M. Junger, L. J. Nieuwenhuis, and R. Joosten, "Why would we get attacked? An analysis of attacker's aims behind DDoS attacks," J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl., vol. 11, no. 2, pp. 3–22, 2020.

[5] Liu and J. Huang, "DDoS Defense Systems in Large Enterprises: A Comprehensive Review of Adoption, Challenges, and Strategies," J. Artif. Intell. Mach. Learn. Manage., vol. 2, no. 1, pp. 1–21, 2018.

[6] R. Uddin, S. A. Kumar, and V. Chamola, "Denial of Service attacks in Edge computing layers: Taxonomy, Vulnerabilities, Threats and Solutions," Ad Hoc Netw., vol. 138, p. 103322, 2023.

[7] Z. Liu, H. Jin, Y. C. Hu, and M. Bailey, "Practical proactive DDoS-attack mitigation via endpoint-driven in-network traffic control," IEEE/ACM Trans. Netw., vol. 26, no. 4, pp. 1948–1961, 2018.

[8] Kumar, "Emerging Threats in Cybersecurity: A Review Article," Int. J. Appl. Nat. Sci., vol. 1, no. 1, pp. 1–8, 2023.

[9] R. R. Brooks, L. Yu, I. Özcelik, J. Oakley, and N. Tusing, "Distributed denial of service (DDoS): a history," IEEE Ann. Hist. Comput., vol. 44, no. 2, pp. 44–54, 2021.

[10] A. Ophardt, "Cyber warfare and the crime of aggression: The need for individual accountability on tomorrow's battlefield," Duke L. Tech. Rev., vol. 9, pp. 1–10, 2010.

[11] Sonar and H. Upadhyay, "A survey: DDOS attack on Internet of Things," Int. J. Eng. Res. Dev., vol. 10, no. 11, pp. 58–63, 2014.

[12] Arora, K. Kumar, and M. Sachdeva, "Impact analysis of recent DDoS attacks," Int. J. Comput. Sci. Eng., vol. 3, no. 2, pp. 877–883, 2011.

[13] H. A. Salman and A. Alsajri, “The Evolution of Cybersecurity Threats and Strategies for Effective Protection. A review”, SHIFRA, vol. 2023, pp. 73–85, Aug. 2023, doi: 10.70470/SHIFRA/2023/009.
[14] Kashaf, V. Sekar, and Y. Agarwal, "Analyzing third party service dependencies in modern web services: Have we learned from the mirai-dyn incident?," in Proc. ACM Internet Meas. Conf., pp. 634–647, Oct. 2020.
[15] R. Singh, S. Tanwar, and T. P. Sharma, "Utilization of blockchain for mitigating the distributed denial of service attacks," Secur. Privacy, vol. 3, no. 3, p. e96, 2020.
[16] V. R. Guntamukalla, "Mitigation Against Distributed-Denial of Service Attacks Using Distribution and Self-Learning Aegis System," Ph.D. dissertation, Texas A&M Univ.-Kingsville, 2017.
[17] Coburn, E. Leverett, and G. Woo, Solving cyber risk: protecting your company and society, John Wiley & Sons, 2018.
[18] İ. Özçelik and R. Brooks, Distributed denial of service attacks: Real-world detection and mitigation, CRC Press, 2020.
[19] Y. L. Khaleel, M. A. Habeeb, and H. Alnabulsi, “Adversarial Attacks in Machine Learning: Key Insights and Defense Approaches ”, Applied Data Science and Analysis, vol. 2024, pp. 121–147, Aug. 2024.
[20] Fachkha, E. Bou-Harb, and M. Debbabi, "Inferring distributed reflection denial of service attacks from darknet," Comput. Commun., vol. 62, pp. 59–71, 2015.
[21] Karami, Y. Park, and D. McCoy, "Stress testing the booters: Understanding and undermining the business of DDoS services," in Proc. 25th Int. Conf. World Wide Web, pp. 1033–1043, Apr. 2016.
[22] J. Scott Sr and W. Summit, "Rise of the machines: The dyn attack was just a practice run," Inst. Crit. Infrastruct. Technol., Washington, DC, USA, 2016.
[23] L. Hussain, “Fortifying AI Against Cyber Threats Advancing Resilient Systems to Combat Adversarial Attacks”, EDRAAK, vol. 2024, pp. 26–31, Mar. 2024, doi: 10.70470/EDRAAK/2024/004.
[24] R. A. Yusof, N. I. Udzir, and A. Selamat, "Systematic literature review and taxonomy for DDoS attack detection and prediction," Int. J. Digit. Enterp. Technol., vol. 1, no. 3, pp. 292–315, 2019.
[25] Z. Gavric and D. Simic, "Overview of DOS attacks on wireless sensor networks and experimental results for simulation of interference attacks," Ingeniería e Investigación, vol. 38, no. 1, pp. 130–138, 2018.
[26] C. S. Kalutharage, X. Liu, C. Chrysoulas, N. Pitropakis, and P. Papadopoulos, "Explainable AI-based DDOS attack identification method for IoT networks," Computers, vol. 12, no. 2, p. 32, 2023.
[27] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," ACM SIGCOMM Comput. Commun. Rev., vol. 34, no. 2, pp. 39–53, 2004.
[28] Akinwale, E. Olajubu, and A. Aderounmu, "A Regeneration Model for Mitigation Against Attacks on HTTP Servers for Mobile Wireless Networks," Int. J. Electr. Comput. Eng. Syst., vol. 15, no. 5, pp. 395–406, 2024.
[29] A. Dogra and N. Taqdir, "Enhancing DDoS Attack Detection and Network Resilience Through Ensemble-Based Packet Processing and Bandwidth Optimization," Deleted J., vol. 2, no. 4, pp. 930–937, 2024. doi: 10.47392/irjaeh.2024.0130.
[30] Tedyyana, O. Ghazali, and O. W. Purbo, "Machine learning for network defense: automated DDoS detection with telegram notification integration," Indones. J. Electr. Eng. Comput. Sci., vol. 34, no. 2, pp. 1102, 2024.
[31] Bindu, A. V. S. Harika, D. Swetha, and M. Sahithi, "SDN Network DDOS Detection Using ML," Int. J. Innov. Sci. Res. Technol., pp. 811–817, 2024.
[32] S. salman Qasim and S. M. NSAIF, “Advancements in Time Series-Based Detection Systems for Distributed Denial-of-Service (DDoS) Attacks: A Comprehensive Review”, BJN, vol. 2024, pp. 9–17, Jan. 2024.
[33] Laiq, F. Al-Obeidat, A. Amin, and F. Moreira, "DDoS Attack Detection in Edge-IIoT Network Using Ensemble Learning," J. Phys. Complex., 2024.
[34] L. Becerra-Suarez, I. Fernández-Roman, and M. G. Forero, "Improvement of Distributed Denial of Service Attack Detection through Machine Learning and Data Processing," Mathematics, vol. 12, no. 9, p. 1294, 2024. doi: 10.3390/math12091294.
[35] Cynthia, D. Ghosh, and G. K. Kamath, "Detection of DDOS attacks using SHAP-Based feature reduction," Int. J. Mach. Learn., vol. 13, no. 4, pp. 173–180, 2023. doi: 10.18178/ijml.2023.13.4.1147.
[36] Z. Zhou, Ensemble Methods: Foundations and Algorithms, CRC Press, 2012. doi: 10.1201/b12207.