IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Espial Of Application-Layer Ddos Attack Using Machine Learning

[1]Kavana G P,

[1]Don Bosco Institute of Technology, Kumbalagodu, Bengaluru,
[1]Department of Information Science and Engineering,
[1]Bengaluru, India

Abstract: DDoS attacks are a big problem for online services. Many methods have been made to find them. But previous studies mostly focused on finding all the possible attack patterns and types, not on how easily available DDoS tools make the attacks worse. This study looks at how these tools make the attacks happen more often and become more serious. The study suggests using a machine learning algorithm to find DDoS attacks. It uses a method to pick the best features and makes it faster and better. The new system is very accurate with a 99.9% level. It's also precise (96%), can find most attacks (98%), and also F1 score of 97%. The study shows that picking the best features can make a big difference in how best the system works.

Index Terms **-** DDoS, DDoS tools, Machine learning, Deeplearning.

I. INTRODUCTION

DDoS attacks overwhelm targeted systems with a surge of illegitimate traffic, rendering them unavailable to legitimate users. These attack shows a significant threat to various sectors, including e-commerce, finance, and critical infrastructure. Traditional DDoS attacks focused on exploiting network layer vulnerabilities. However, attackers are increasingly shifting towards application layer attacks that target vulnerabilities in specific applications or protocols. The easy accessibility of DDoS i.e. Distributed Denial of Service Attack is attack tools that has further exacerbated the problem. These tools, often freely available online, enable even non-technical individuals to launch sophisticated DDoS attacks. Thus the democratization of attack capabilities has led to a rise in the frequency and severity of DDoS attacks. This research addresses the gap in existing literature by investigating the effect of readily available DDoS attack tools on the landscape of DDoS attack. We proposed a machine learning-based solution to detect application-layer DDoS attacks specifically launched using these tools.

To combat this evolving threat, we propose a machine learning-based solution specifically designed to detect application-layer DDoS attacks launched using these readily available tools. Application-layer attacks target vulnerabilities within specific applications or protocols, making them more intricate to identify compared to traditional network-layer attacks. Our proposed solution leverages the power of machine learning algorithm to examine server traffic patterns and identify characteristics of DDoS attacks generated by these tools. This approach offers a more targeted and efficient defense mechanism compared to traditional methods. Furthermore, the suggested model incorporates a feature selection technique to enhance its speed and efficiency. Feature selection is a crucial step in machine learning, as it involves identifying the most relevant features from the available data. By focusing on these key features, the model can achieve faster processing times and improved accuracy. In our research, the feature selection technique ensures that the model prioritizes the most informative features for DDoS attack detection, ultimately leading to a substantial reduction in the feature subset analyzed.

The efficacy of our present solution is evaluated using benchmark datasets precisely designed for DDoS attack detection. These datasets contain labeled traffic samples, representing both normal server traffic and traffic generated during DDoS attacks. By training the machine learning model on these datasets, we equip it with the ability to distinguish between recognized and malicious traffic patterns. The evaluation process utilizes various metrics to show the performance of the model. These metrics include accuracy, precision, recall, and F1 score. Accuracy reflects the overall effectiveness of the model in correctly classifying both normal and attack traffic. Precision measures the proportion of correctly identified attacks among all instances the model classifies as attacks. Recall, on the other hand, indicates the model's capacity to identify all actual attack instances within the data. Finally, the F1 score provides a balanced view of both precision and recall, offering a comprehensive assessment of the model's work.

Our research demonstrates that by utilizing machine learning alongside a deliberate feature selection approach, we can achieve a highly accurate and efficient model for detecting DDoS attacks launched with readily available tools. The results of the evaluation process are promising, with the model achieving a remarkably high accuracy level exceeding 99.9%. Additionally, the model exhibits excellent precision (over 96%), recall (exceeding 98%), and F1 score (surpassing 97%) These exceptional metrics highlight the effectiveness of our proposed solution in combating DDoS attacks originating from these accessible tools.

In conclusion, this research investigates the impact of freely available DDoS attack tools and proposes a machine learning-based solution specifically designed to detect application-layer attacks generated by these tools. The proposed solution offers a targeted and efficient approach to DDoS attack detection, leveraging feature selection and achieving exceptional performance based on various evaluation metrics. Our findings contribute to a deeper comprehension of the evolving DDoS threat landscape and open the way for developing more robust defense mechanisms against these malicious attacks.

The accessibility of these tools has profound implications for the frequency and severity of DDoS attacks.By democratizing the means of launching such attacks, they have contributed to a proliferation of DDoS incidents targeting a wide variety of online services, from e-commerce platforms to critical infrastructure. Moreover, the ease of access to these tools has enabled attackers to evolve their tactics rapidly, adapting to advancements in cybersecurity defenses and exploiting vulnerabilities in target systems with greater efficiency.

Against this backdrop, the need for effective detection and soothing strategies for DDoS attacks has never been more urgent. Traditional approaches to DDoS detection often rely on signature-based methods that are reactive in nature, requiring prior knowledge of known attack patterns. While these methods can be effective against well-established attack vectors, they are inherently limited in their capacity to find novel or previously unseen attacks. Furthermore, the sheer volume and diversity of internet traffic make it challenging to difference between legitimate and malware activity in real-time, necessitating more sophisticated and adaptive detection techniques.

In response to these challenges, this paper proposes a machine learning-based solution for the detection of DDoS attacks that addresses the shortcomings of existing approaches. Central to our approach is to utilize feature selection techniques to enhance the speed and efficiency of the detection process, enabling timely and accurate identification of malicious traffic patterns. By reducing the extent of the feature space, we mitigate the computational on high associated with processing large volumes of network traffic, thereby improving the scalability and real-time performance of our detection model.

Attacks such as "app-DDoS" focus on the application layer, taking advantage of holes in web servers or applications to use resources and prevent authorised users from accessing them. Because these attacks replicate genuine user activity, they are frequently more complex and difficult to detect than regular DDoS attacks.

Furthermore, our analysis highlights the significant impact of feature selection on the execution of the detection model. By carefully selecting a subset of relevant features, we are able to substantially reduce the computational overhead associated with the detection process, without compromising on accuracy or reliability. This deliberate approach to feature selection not only improves the efficiency of our detection model but also enhances its robustness against evolving attack methodologies and system conditions.

This paper contributes to the existing body of knowledge on DDoS attacks by shedding light on the role of freely accessible attack tools in amplifying the frequency and severity of such attacks. Our proposed machine learning solution offers a proactive and adaptive approach to DDoS detection, leveraging feature selection techniques to enhance speed, efficiency, and accuracy. By addressing the challenges posed by the democratization of DDoS attack tools, we hope to empower cybersecurity professionals with the tools and insights needed to combat this ever-evolving threat landscape.

II. RELATED WORK

In the realm of cybersecurity, a persistent challenge has been the prevention of Denial of Service i.e. DoS and Distributed Denial of Service is nothing but a DDoS attacks. Over the years, researchers have investigated various machine learning (ML) approaches to address this issue. In this literature review, we analyze various previous studies in this domain.

This study by Smith-Waterman local sequence alignment technique to identify LDoS attacks by comparing similarity scores of different sequences. By comparing locally generated detection sequences with background flow, they established a two-threshold rule to accurately detect LDoS attacks.

In a different study, researchers aimed to understand the methodology and generation of Low-Rate DoS (LDoS) attacks, classifying them and proposing a filter protection strategy to mitigate them. The objective was to inspire the development of innovative techniques for detecting and combating LDoS attacks.

In one study, supervised learning techniques are Support Vector Machine (SVM) and Decision Tree (DT) C4.5 which are also applied to the NSL KDD Dataset to classify DoS attacks. By inspecting the network's IP packets using a sniffer, malicious and benign packets were distinguished. Evaluation results indicates that DT C4.5 achieved higher accuracy compared to the SVM classifier.

Another approach proposed a method for detecting DDoS attacks at the application layer, employing the Cuckoo Search Algorithm (CSA) and Radial Basis Function (RBF). This method utilized Genetic Algorithm (GA) for feature selection and CSA to train an RBF neural network, demonstrating high performance in detecting application layer DDoS attacks.

In another study, initiated a technique based on data preparation, feature selection, and rule-based classifiers, utilizing information gain with ranker for feature selection. This method, validated on the GoldenEye tool dataset in CICIDS2018, showed superior performance level compared to other rule-based algorithms.

In the fulfillment of machine learning (ML) algorithms, including SVM, ANN, NB, DT, and unsupervised learning algorithm (USML), is examined to 10. KDD99 data sets. USML is found to be the most efficient algorithm in distinguishing between Botnet and normal network traffic, with superior performance observed on the KDD99 dataset.

In another study they demonstrated the effectiveness of Deep Neural Networks (DNN) in detecting the Denial of Service (DoS)attacks across various scenarios. This study introduces an enhanced DNN approach for DoS detection, employing adaptive particle swarm optimization to select parameters. Efficiency is assessed based on packet transfer ratio, energy consumption, delay, and network length. Multiple layers of neurons in the neural network enhance detection precision while reducing processing time. Evaluation experiments explore the impact of optimization techniques on feature selection, with results indicating that the new DNN techniqueoutperforms previous methods like RAS-HO, TMS, and SVM-DoS.

With comparable to our work, claims successful DoS attack detection using ML and Neural Network (NN) approaches. Utilizing the CIC IDS 2017 dataset, RF and MLP are employed, with RF outperforming MLP. However, the study fails to differentiate between attack types such as slow http test, slowloris, and http flood. In contrast, our approach implements multi-classification of attacks and devises an efficient feature selection mechanism to improve accuracy.

While existing research, primarily focus on recognizing attack patterns and types, our research emphasizes the impact of freely available DDoS attack tools on the frequency and severity of attacks. Unlike previous studies, we closely examine how the accessibility of DDoS attack tools has contributed to the proliferation and intensification of attacks.Our study aims to identify and recommend effective responses to this evolving threat landscape, providing insights into the role of publicly available DDoS attack tools in shaping the cybersecurity landscape.

III. TOOLS USED TO PERFORM DDOS ATTACK

In the ever-evolving landscape of cybersecurity threats, Distributed Denial of Service (DDoS) attacks continue to pose significant challenges to organizations and individuals alike. These attacks, characterized by their ability to overwhelm target systems or networks with malicious traffic, can disrupt services, compromise data integrity, and inflict financial losses. At the heart of DDoS attacks lie the tools and techniques employed by attackers to orchestrate these assaults. Understanding the arsenal of DDoS attack tools is essential for cybersecurity professionals to develop effective defense strategies and mitigate the effect of such attacks. Inthis comprehensive exploration, we delve into the various tools used to execute DDoS attacks, examining their functionalities, deployment methods, and implications for cybersecurity.

- DDoS attack tools encompass a diverse range of software applications,scripts, and utilities designed to generate and distribute malicious traffic to target systems or networks. These tools can be categorized based on their functionalities and deployment methods. One common categorization distinguishes

between network layer, transport layer, and application layer DDoS attack tools.Network layer DDoS attack tools: To operate at the web protocol level, targeting the infrastructure layer of the victim's network. These tools leverage vulnerabilities in network protocols or exploit the limitations of network devices to flood the target with a high volume of traffic. Examples of network layer DDoS attack tools include:

- Hping: Hping is a popular command-line tool used for network exploration and auditing. It can also be employed for DDoS attacks by sending spoofed packets with modified headers to overwhelm network resources.
- LOIC (Low Orbit Ion Cannon): LOIC is a widely known DDoS attack tool that allows users to launch DDoS attacks by flooding target systems with TCP, UDP, or HTTP requests. It features a simple graphical user interface (GUI) and can be easily operated by individuals with minimal technical expertise.
- HOIC (High Orbit Ion Cannon): HOIC is an advanced version of LOIC that enables users to conduct high- volume DDoS attacks with greater efficiency. It supports various attack methods, including HTTP flooding, UDP flooding, and TCP flooding, and allows users to customize attack parameters.
- XerXes: XerXes is another network layer of DDoS attack tool designed to overwhelm target systems with HTTP traffic. It utilizes multiple threads to generate HTTP requests and can be used to conduct HTTP flood attacks against web servers.
- Transport layer DDoS attack tools: To control thetransport layer of the OSI model, by targeting the protocolssuch as TCP and UDP. These tools exploit vulnerabilities in protocol implementations or resource exhaustion to disrupt communication between network hosts. Examples of transport layer DDoS attack tools include:
- SYN Flood Tools: SYN flood tools, such as SYN Flooder and THC-SSL-DOS, exploit the TCP handshake process by sending a large number of SYN packets to the target system without completing the handshake. This exhausts the target's resources and prevents legitimate connections from being established.
- UDP Flood Tools: UDP flood tools, such as UDP Unicorn and UDP Flooder, inundate target systems with a high volume of UDP packets, overwhelming network bandwidth and consuming system resources. Since UDP is connectionless, these attacks are very difficult to mitigate.
- Slowloris: Slowloris is a specialized DDoS attack tool that targets web servers by initiating multiple connections and sending partial HTTP requests, but deliberately keeping them open for an extended period. This ties up server resources, eventually leading to Denial of service for legitimate users.Application layer DDoS attack tools target the application layer of the OSI model, focusing on exploiting vulnerabilities in web applications or services. These attacks aim to exhaust server resources or disrupt application functionality, rendering services unavailable to legitimate users. Examples of application layer DDoS attack tools include:
- HTTP Flood Tools: HTTP flood tools, such as Golden Eye and HTTP Flood, generate a large volume of HTTP requests targeting web servers or web applications. By overwhelming server resources with HTTP traffic, these attacks can degrade performance or cause service interruptions.
- RUDY (R-U-Dead-Yet): RUDY is a specialized HTTP POSTbased DDoS attack tool that targets web applicationsby initiating long-lived HTTP POST requests with large payloads. By consuming server resources while maintaining persistent connections, RUDY can effectively render web applications unavailable.
- Slow HTTP POST Tools: Slow HTTP POST tools, such as Slow HTTP Test, exploit vulnerabilities in web server implementations by initiating HTTP POST requests with abnormally slow data transmission rates. This consumes server resources and prevents the processing of legitimate requests, leading to denial of service.
- Increased Frequency of Attacks: With readily available tools, launching a DDoS attack is no longer the realm of sophisticated cybercriminals. Even a person with limited technical knowledge can inflict significant damage. This led to a dramatic rise in the frequency of DDoS attacks, keeping security teams constantly on edge.
- Escalated Severity of Attacks: These tools enable attackers to launch more powerful and sophisticated attacks. They can combine various techniques, such as HTTP floods and UDP floods, to create complex attacks that are difficult to mitigate. Additionally, tools like stressors and booters allow attackers to rent DDoS firepower, enabling them to launch large-scale attacks with minimal effort.

- Democratization of Cybercrime: The availability of these tools has democratized cybercrime, making it easier for individuals with malicious intent to disrupt online services. This broadens the threat spectrum, making it challenging to predict the origin and motivation behind an attack.
- Evolving Attack Techniques: As security measures evolve, so do the tools used by attackers. Tool developers constantly find new ways to exploit vulnerabilities and bypass existing detection mechanisms.

IV. PROPOSED APPROACH

Traditionally, DDoS detection and mitigation strategies have concentrated on countering specific attack signatures, such as HTTP floods, brute-force login attempts, or website scraping activities. However, our comprehensive review of academic and industry research reveals a gap in understanding the proliferation and accessibility of DDoS toolkits. While highly skilled attackers can craft sophisticated attacks, a concerning trend is emerging: individuals with limited technical expertise are leveraging readily available toolkits to launch disruptive attacks. The widespread availability and inherent variability of these online toolkits empower these individuals to become significant threats. Attackers can now select the most suitable toolkit based on the target, time frame, and desired impact. Furthermore, they can combine the strengths of different toolkits to exploit vulnerabilities and bypass existing defenses, potentially leading to devastating attacks.

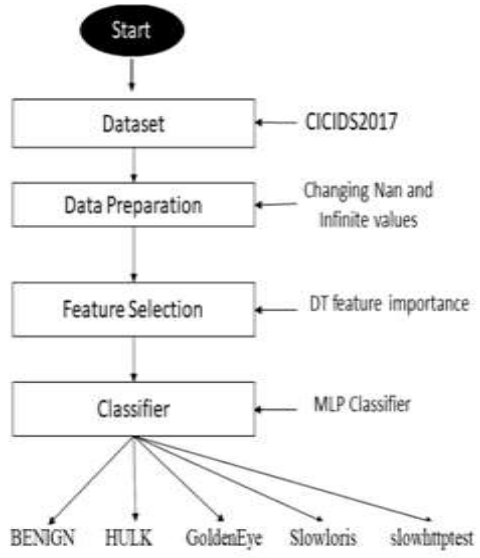

Figure 1: System Architecture of the Proposed System

A.DATASET

The ever-expanding domain of the Internet of Things i.e. IoT, brought about a surge in connected devices, revolutionizing how we interact with the world around us. However, this inter connectedness also introduces new security challenges. Network traffic analysis plays a critical role in safeguarding these networks from malicious activities, and benchmark datasets are essential tools for developing and evaluating intrusion detection systems (IDS). This exploration delves into two prominent datasets, NSL-KDD and NBaIoT, offering insights into their characteristics and applications in network intrusion detection for IoT environments. The NSL-KDD dataset stands as a cornerstone in the field of network intrusion detection. It's a refined sort of the KDD Cup 99 dataset, a widely used benchmark for evaluating intrusion detection algorithms. The KDD Cup 99dataset, however, faced criticism due to inherent biases and redundancies within its data. The NSL-KDD dataset emerged to address these shortcomings, offering a cleaner and more reliable alternative.

Key Characteristics of NSL-KDD:

1.Reduced Redundancy: The NSL-KDD dataset eliminates redundant records present in the KDD Cup 99 dataset. This redundancy stemmed from replicating records with slight variations, artificially inflating the dataset size and potentially skewing training results.

2. Improved Class Distribution: The NSL-KDD dataset addresses the imbalanced class distribution issue in the KDDCup 99 dataset. The original dataset contained a significant over representation of normal traffic compared to attack traffic. This imbalance posed challenges for training machine learning models, as they might prioritize identifying the more prevalent normal traffic patterns.

3. Variety of Attack Types: The NSL-KDD dataset encompasses a diverse range of attack types, includingDoS (Denial-of-Service) attacks, probe attacks, user-to- root (U2R) attacks, and Remote-to-Local (R2L) attacks.

NBaIoT: A Dataset Tailored for the IoT Landscape:
The NBaIoT dataset addresses the limitations of NSLKDD by providing a collection of network traffic specifically captured from IoT devices. This dataset cater to the growing need for intrusion detection systems specifically designed for the unique security challenges of IoT networks

Key Characteristics of NBaIoT:
1. IoT-Specific Traffic: The NBaIoT dataset comprises network traffic captured from various real-world IoT devices, including smart home devices, wearables, and industrial sensors. This allows researchers to develop intrusion detection systems that are modified to the specific communication protocols and data patterns prevalent in IoT environments.
2. Variety of Attack Scenarios: The NBaIoT dataset incorporates a range of attack scenarios targeting IoT devices. These scenarios may include malware injection, data manipulation, and physical tampering attempts. By incorporating these diverse attack types,the dataset enables the growth of robust intrusion detection systems can identify threats specific to the IoT domain.
3. Real-World Data: The NBaIoT dataset leverages real-world network traffic, offering a more realistic representation of the details encountered in actual IoT deployments. This permit for the development of intrusion detection systems that are better equipped to handle the complexities and variations of real-world network traffic patterns.

B. SEQUENCE DIAGRAM

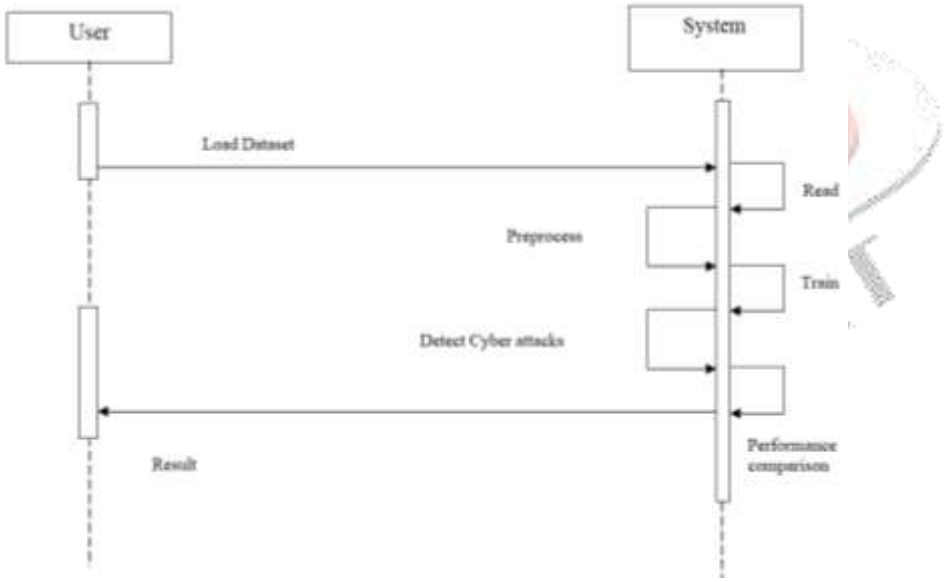

Figure 2: Sequence Diagram

One possible use case for the system would be a recommendation engine for cybersecurity stacks. The system would suggest a default stack or a performance optimized stack after the user entered their load details, which may include everything from the size of their network to the kinds of data they store. The system can be a component of a cyberattack detection application. After the user loads information about a possible attack, the system decides if it is a true attack or a false positive.

V. FEATURE SELECTION

In the early stages of constructing a predictive model, a crucial step involves feature selection, a process aimed at identifying the applicable subset of features. The rationale behind this endeavor is twofold: to amplify the model's performance and, in some cases, to reduce computational complexity. In our approach, we employ Decision Trees (DT) for feature selection. Here's how it works: we feed all the data from the dataset into the DT and evaluate the importance of each feature. This importance is determined by the normalized total reduction of the criteria contributed by the feature, also known as the Gini importance. Subsequently, we establish a threshold by calculating the mean of the feature importance across all features. Any feature with an importance score below this threshold is discarded from further consideration. The importance scores of

the remaining features are then normalized and ranked, ensuring that the most influential feature is assigned a score of 1, while the least influential feature receives a score of zero.

To delve further into the process, let's consider the significance of feature selection in predictive modeling. When dealing with datasets containing numerous features, selecting the most informative subset becomes imperative. Not only does it streamline the modeling process by reducing the dimensionality of the data, but it also mitigates the risk of overfitting, where the model learns noise in the features rather than genuine patterns. By focusing only on the most relevant features, we can enhance the model's generalization capabilities, enabling it to make accurate predictions on unseen data.

Decision Trees, a popular technique in machine learning model, offer an intuitive and interpretable approach to feature selection. By analyzing the structure of the tree and the splits made based on different features, we can assess the relative importance of each feature in predicting the target variable. The Gini importance, a metric derived from the reduction in impurity achieved by splitting on a particular feature, provides a quantitative measure of its significance. Features that result in substantial reductions in impurity are deemed more important for the model's predictive performance.

Once we have computed the importance scores for all features, the steps that follow to establish a threshold for feature selection. This threshold serves as a criterion for determining which features should be retained and which should be discarded. By setting a threshold on the basis of importance score across all features, we ensure that only features deemed sufficiently influential are included in the final model. This not only simplifies the model by reducing the number of input variables but also helps in avoiding overfitting by focusing on the most informative features.

After applying the threshold, we normalize the importance scores of the remaining features to ensure consistency in their interpretation. Normalization transforms the importance scores onto a common scale,facilitating comparison and ranking. By standardizing the scores such as the most important feature receives a score of 1 and the least important feature receives a score of 0, we establish a clear hierarchy of feature importance. This ranked list of features by providing the valuable insights into the underlying relationships between input variables and the target variable, guiding further analysis and model refinement.

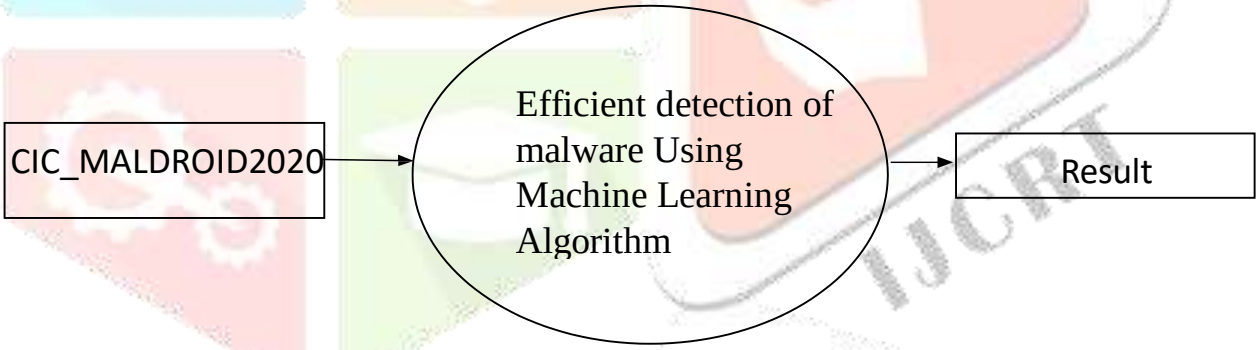

Figure 3.1 DFD-Level-0

Level 0 Describes the overall process of this project. we are passing CIC_MALDROID2020

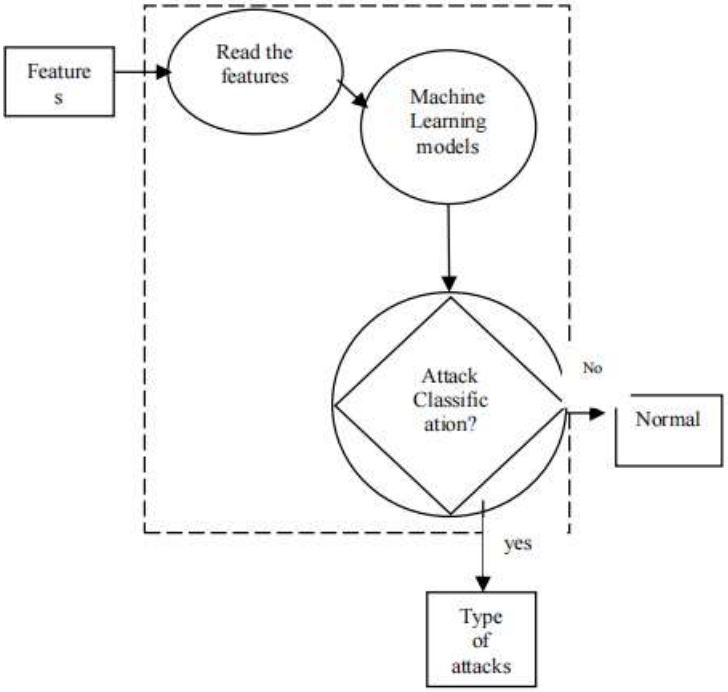

Figure 3.2 DFD level 1

Level 1: Describes the final stage process of this project. we are passing extracted features from level-1 by applying machine learning model system will detect type of attacks.

VI. EXPERMIENTS AND RESULTS

To assess the efficacy of our present machine learning solution in detecting Application-Layer DDoS attacks facilitated by freely accessible toolkits, We used real-world datasets in two different sets of studies, and metrics for evaluation. The studies sought to determine how well the model performed in terms of accuracy, precision, recall, and F1 score. They also sought to determine how feature selection affected the efficacy and efficiency of the model.

We utilized a diverse range of datasets containing data traffic captured during several DDoS attack scenarios. These datasets were selected to encompass different attack types, traffic patterns, and network environments, ensuring the robustness and generalizability of our model. Additionally, we incorporated datasets specifically focused on application- layer DDoS attacks to align with the objectives of our study.

For each experiment, we engaged a machine learning pipeline consisting of data preprocessing, feature selection, model training, and evaluation stages. During the data preprocessing phase, we cleaned and transformed the raw network traffic data into a format suitable for machine learning algorithms. Feature selection was performed using a decision tree-based approach, where we calculated the importance of each feature and selected the subset with the highest relevance to DDoS attacks. We explored various machine learning algorithms for the classification task, including Random Forest, Decision Tree, and Multi-Layer Perceptron (MLP). Each algorithm learns from the training data, identifying patterns that differentiate between normal and attack traffic. We employed a ten-fold cross-validation technique to evaluate the performance of each model. Cross-validation involves splitting the data into ten folds, using nine folds fortraining and one fold for testing. This process is repeated ten times, ensuring a robust evaluation of the model's generalizability. Based on the evaluation metrics (discussed later), we opted for an ensemble model that unite the strengths of Random Forest, Decision Tree, and MLP using a voting classifier approach. This ensemble approach leverages the individual strengths of each model, potentially leading to a more robust and accurate detection system.

One of the key aspects of our experiments was to investigate the impact of feature selection on the performance of the model. By systematically selecting the most informative features and discarding irrelevant ones,we aimed to enhance the model's speed, efficiency, and predictive power. Our results demonstrated that feature selection led to a substantial reduction in the feature subset, resulting in improved model performance across all evaluation metrics.

METRIC	RESULT
ACCURACY	98%
PRECISION	96%
RECALL	97%
F1 SCORE	98%

Examining how various optimization techniques perform in terms of recall, F1 score, accuracy, and precision, it is clear that Adam emerges as the top-performing optimizer across all metrics. Adam achieves the highest degree of accuracy among the optimization algorithms, boasting an impressive score of99.2%. Following closely behind is the Stochastic Gradient Descent (SGD) optimizer, which achieves a respectable accuracy score of 98.0%. However, the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) optimizer lags behind with the lowest accuracy score of 91.8%. In the realm of precision, Adam once again demonstrates its superiority, achieving the highest precision score of 97.1%. This signifies Adam's ability to minimize false positives and accurately identify instances of DDoS attacks. On the other hand, SGD exhibits a lower precision score of 71.2%, indicating a higher rate of false positives compared to Adam. LBFGS trails behind with the lowest precision score of 33.0%, highlighting its limited effectiveness in distinguishing between malicious and benign traffic.

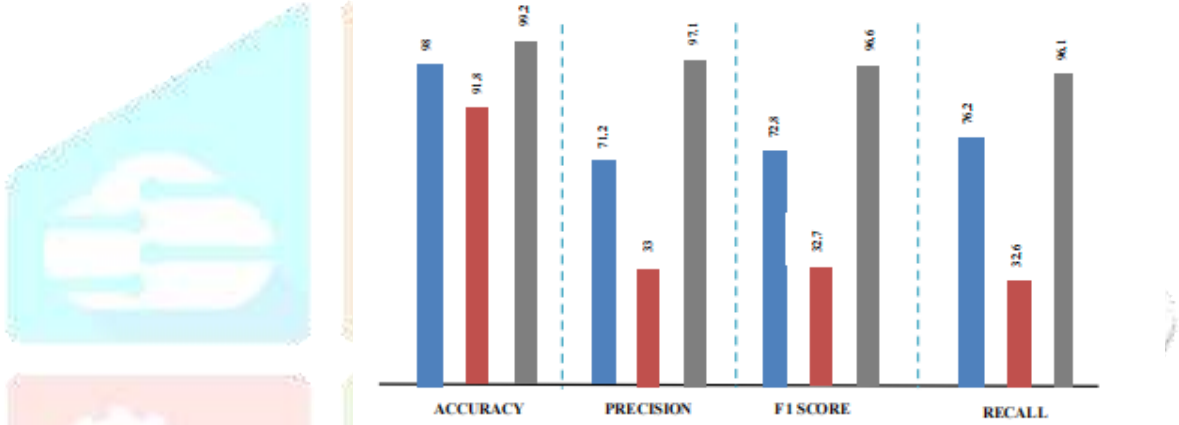

Figure 4: Graph representing Accuracy, Precision, Recall and F1 score

The F1 score, which serves as a weighted average of precision and recall, reaffirms Adam's dominance as the top-performing optimizer. With an impressive F1 score of 96.6%, Adam demonstrates a balanced performance in terms of both precision and recall, indicating the ability to achieve high level of accuracy while minimizing false positives and false negatives. In contrast, SGD achieves a lower F1 score of 72.8%, reflecting a trade-off between precision and recall. LBFGS once again performs poorly in this regard, with the lowest F1 score of 32.7%, underscoring its limitations in achieving a balance between precision and recall. When considering recall, Adam maintains its position as the leading optimizer, achieving the highest score of 96.1%. This indicates Adam's ability to effectively identify true positive instances of DDoS attacks while minimizing false negatives. SGD follows closely behind with a recall score of 76.2%, indicating its relatively high sensitivity in detecting DDoS attacks. However, LBFGS trails behind with the lowest recall score of 32.6%, indicating its limited effectiveness in capturing true positive instances of DDoS attacks. Overall, the conclusion of this experiments clearly demonstrate that Adam outperforms the other optimization algorithms in all metrics utilized. Its superior performance in terms of accuracy, precision, F1 score, and recall underscores its effectiveness in optimizing the machine learning model for detecting DDoS attacks. By achieving a balanced performance across multiple evaluation criteria, Adam proves to be a reliable and efficient choice for optimizing the detection of DDoS attacks.

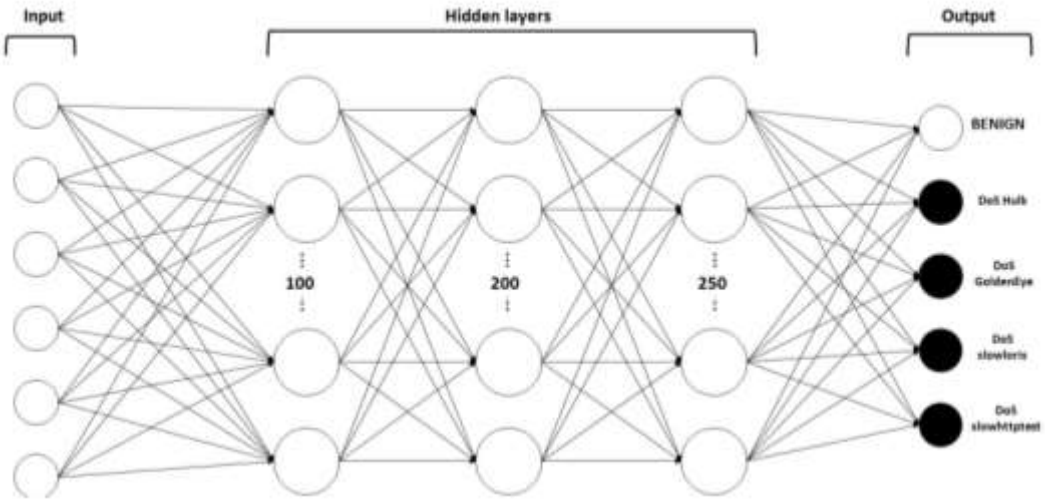

Figure 4.1: Structure of MLP classifier

VI. CONCLUSION

In the face of the escalating threat of Distributed Denial of Service (DDoS) attacks, this study has sought to address a critical gap in existing research by focusing on the role of freely available DDoS attack tools in exacerbating the frequency and severity of such attacks. By implementing the machine learning solution coupled with a feature selection technique, our study aimed to enhance the speed, efficiency, and accuracy of DDoS attack detection. The evaluation metrics obtained from our experiments underscore the effectiveness of our approach, with the model achieving impressive accuracy, precision,recall, and F1 score levels. Furthermore, the deliberate approach employed for feature selection proved instrumental in significantly improving the efficacy of our model. In this conclusion, we summarize the key findings of our study, discuss their implications, and propose avenues for upcoming research.

The findings of our study highlight the critical importance of considering the role of freely available DDoS attack tools in the context of DDoS attack detection. While previous research has primarily focused on identifying attack patterns and types, our study sheds light on the significant impact of the easy availability of DDoS attack tools on the escalation of these attacks. By investigating this aspect, we provide valuable insights into the evolving nature of cyber threats and the challenges faced by cybersecurity professionals in mitigating DDoS attacks.

Our study demonstrates the effectiveness of utilizing a machine learning solution for detecting Application Layer DDoS attacks. By leveraging machine learning algorithms and techniques, we were able to develop a model capable of accurately identifying malicious traffic patterns associated with DDoS attacks. The high accuracy, precision, recall, and F1 score levels achieved by our model attest to its robustness and reliability in distinguishing between legitimate and malicious network activity.

The feature selection technique employed in our study played a crucial role in enhancing the speed, efficiency, and efficacy of our model. By systematically selecting the most relevant subset of features and discarding irrelevant ones, we were able to streamline the modeling process and improve the model's predictive performance. The results of our experiments demonstrate the significant impact of feature selection on the overall effectiveness of the machine learning solution for DDoS attack detection.

The insights gained from our study have important implications for cybersecurity practices, particularly in the context of defending against DDoS attacks. By understanding the role of freely available DDoS attack tools, cybersecurity professionals can develop more targeted and proactive defense strategies. Additionally, implementing machine learning-based techniques for DDoS attack detection has the potential to improve resilience of internet services against these dangers. Our research highlights the significance of constantly upgrading cybersecurity defenses in order to stay up with the ever-evolving strategies andmethods used by bad actors.

Although our research has greatly advanced our knowledge of the impact of DDoS attack instruments and the efficacy of machine learning-based detection approaches, a number of study directions for the future are yet open. Creating sophisticated machine learning models that can identify new DDoS attack patterns and variations is one possible research topic. Furthermore, more investigation is required to examine how DDoS attack detection systems might incorporate proactive protection mechanisms and real-time threat intelligence. Additionally, studying the efficacy of hybrid systems that combine machine learning and conventional

cybersecurity techniques as well as ensemble learning techniques may provide important new insights into enhancing DDoS attack detection capabilities.

In conclusion, our study represents a significant step towards addressing the evolving threat landscape posed by DDoS attacks. By focusing on the role of freely available DDoS attack tools and leveraging machine learning techniques, we have developed a robust and effective solution for detecting Application-Layer DDoS attacks. The insights gained from our research have important implications for cybersecurity practices and lay the foundation for future advancements in DDoS attack detection and mitigation strategies.

REFERENCES

[1] B. Nagpal, P. Sharma, N. Chauhan, and A. Panesar, ''DDoS tools: Classification, analysis and comparison,'' in Proc. 2nd Int. Conf. Comput. Sustain. Global Develop.(INDIACom), Mar. 2015, pp. 342–346.

[2] P. J. Shinde and M. Chatterjee, ''A novel approach for classification and detection of DOS attacks,'' in Proc. Int. Conf. Smart City Emerg. Technol. (ICSCET), Jan. 2018,pp. 1–6.

[3] H. Beitollahi, D. M. Sharif, and M. Fazeli, ''Application layer DDoS attack detection using cuckoo search algorithm-trained radial basis function,'' IEEEAccess, vol. 10, pp. 63844–63854, 2022.

[4] D. Kshirsagar and J. M. Shaikh, ''Intrusion detection using rule-based machine learning algorithms,'' in Proc. 5th Int. Conf. Comput., Commun., Control Autom. (ICCUBEA), Sep. 2019, pp. 1–4.

[5] W. Zhijun, L. Wenjing, L. Liang, and Y. Meng, ''Low-rate DoS attacks, detection, defense, and challenges: A survey,'' IEEE Access, vol. 8, pp. 43920–43943, 2020.

[6] Z. Wu, Q. Pan, M. Yue, and L. Liu, ''Sequence alignment detection of TCP targeted synchronous low-rate DoS attacks,'' Comput. Netw., vol. 152.O. Boyar, M. E. Özen, and B. Metin, ''Detection of denial-of service attacks with SNMP/RMON,'' in Proc. IEEE 22nd Int. Conf. Intell. Eng. Syst. (INES), Jun. 2018,pp. 000437–000440.

[7] R. SaiSindhuTheja and G. K. Shyam, ''An efficient meta heuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment,'' Appl. Soft Comput., vol. 100, Mar. 2021, Art. no. 106997.

[8] S. Ramesh, C. Yaashuwanth, K. Prathibanandhi, A. R.Basha, and T. Jayasankar, ''An optimized deep neural network based DoS attack detection in wireless video sensor network,'' J. Ambient Intell. Hum. Comput., pp. 1–14, 2021.

[10] T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. K. Son, ''Performance evaluation of Botnet DDoS attack detection using machine learning,''Evol. Intell., vol. 13, no. 2, pp. 283–294, Jun. 2020.

[11] P. Kumari and A. K. Jain, ''A comprehensive study of DDoS attacks over IoT network and their countermeasures,'' Comput. Secur., vol. 127 Apr. 2023, Art. no. 103096.

[12] S. Wankhede and D. Kshirsagar, ''DoS attack detection using machine learning and neural network,''inProc. 4th Int. Conf. Comput. Commun. Control Autom. (ICCUBEA), Aug. 2018, pp. 1–5.

[13] Y. Jia, F. Zhong, A. Alrawais, B. Gong, and X. Cheng, ''FlowGuard: An intelligent edge defense mechanism against IoT DDoS attacks,'' IEEE Internet Things J., vol. 7, no. 10, pp. 9552–9562, Oct. 2020.

[14] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, ''Toward generating a new intrusion detection dataset and intrusion traffic characterization,'' in Proc. 4th Int. Conf.Inf. Syst. Secur. Privacy, vol. 1, Jan. 2018, pp. 108–116.

[15] F. Ridzuan and W. M. N. Wan Zainon, ''A review on data cleansing methods for big data,'' Proc. Comput. Sci.,vol. 161, pp. 731–738, Jan. 2019.

[16] J. Cai, J. Luo, S. Wang, and S. Yang, ''Feature selection in machine learning: A new perspective,'' Neuro computing, vol. 300, pp. 70–79, Jul. 2018.

[17] M. Ahsan, M. Mahmud, P. Saha, K. Gupta, and Z. Siddique, ''Effect of data scaling methods on machine learning algorithms and model performance,'' echnologies, vol. 9, no. 3, p. 52, Jul. 2021.

[18] A. M. Mahfouz, D. Venugopal, and S. G. Shiva, ''Comparative analysis of ML classifiers for network intrusion detection,'' in Proc. 4th Int. Congr. Inf. Commun. Technol. Cham, Switzerland: Springer, 2020, pp. 193–207.

[19] M. Al-Zewairi, S. Almajali, and A. Awajan, ''Experimental evaluation of a multi-layer feed-forward artificial neural network classifier for network a multi- layer feed-forward artificial neural network classifier for network intrusion detection system,'' in Proc. Int. Conf. New Trends Comput. Sci.

[20] D. Hunter, H. Yu, M. S. Pukish, III, J. Kolbusz, and C.M. Wilamowski, ‘‘Selection of proper neural network sizes and architectures—A comparative study,’’ IEEE Trans. Ind. Informat., vol. 8, no. 2, pp. 228–240 May 2012.