Enhancing Intrusion Detection Systems through Adaptive Learning and Deep Learning Integration

Amogh Prabhu

avprabhu_b20@it.vjti.ac.in

Veermata Jijabai Technological Institute

Tushar Mali

Veermata Jijabai Technological Institute

Nirbhay Hanjura

Veermata Jijabai Technological Institute

Soumitra Kand

Veermata Jijabai Technological Institute

Shrinivas Khedkar

Veermata Jijabai Technological Institute

Research Article

Keywords: Adaptive Learning, Intrusion Detection System, Snort, Zeek, Zeek-Flowmeter, Metasploit, Scapy, CICIDS, PCAPs

Posted Date: October 16th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4580867/v1

Additional Declarations: No competing interests reported.

Enhancing Intrusion Detection Systems through Adaptive Learning and Deep Learning Integration

Amogh Prabhu
Department of CE/IT
Veermata Jijabai Technological Institute
Mumbai, India
avprabhu_b20@it.vjti.ac.in

Tushar Mali
Department of CE/IT
Veermata Jijabai Technological Institute
Mumbai, India
trmali_b20@it.vjti.ac.in

Nirbhay Hanjura
Department of CE/IT
Veermata Jijabai Technological Institute
Mumbai, India
nahanjura_b20@it.vjti.ac.in

Soumitra Kand
Department of CE/IT
Veermata Jijabai Technological Institute
Mumbai, India
smkand_b20@it.vjti.ac.in

Shrinivas A. Khedkar
Department of CE/IT)
Veermata Jijabai Technological Institute
Mumbai, India
sakhedkar@ce.vjti.ac.in

I. Abstract

The objective of this paper is to design an adaptive intrusion detection system that combines Signature Based Intrusion Detection System (SNIDS) and machine learning/deep learning techniques to enhance the anomaly detection.The rapid advancement of digital technology has fueled an unprecedented growth in computer networks, which has played a vital role in fostering social and economic development. These networks have become indispensable across critical sectors and within leading multinational corporations. However, the number of security threats targeting computer networks has increased significantly over the past decade, becoming increasingly audacious and pervasive. The networks display significant heterogeneity and are subjected to a relentless stream of diverse and evolving attacks, emphasizing the necessity for an adaptive IDS system capable of adjusting to new and changing network environments.

In this study, we propose a Adaptive hybrid NIDS designed to detect cyber attacks in an heterogeneous attack environment. Our NIDS framework leverages the adaptive learning techniques used along with the signature base IDS and machine learning based IDS to detect unknown attacks. Various machine learning models, including LSTM, CNN, MLP, and Decision Tree are used in this adaptive system. Our hybrid Network intrusion detection model is based on Snort-3 as the SNIDS, which provides high efficiency and better prediction results. The network baseline includes the set of benign traffic patterns and self generated datasets using Metasploit and Scapy. Our study compares the results of models trained on CICIDS2017 dataset and the generated dataset. The generated dataset provides better flexibility and performance in real-time setup due to the use of Zeek-flowmeter instead of CICFlowmeter. To detect real-time dynamic traffic attacks, a dashboard integrated with SNIDS and ML/DL-based IDS is developed. This dashboard is subsequently utilized to assess the Adaptive NIDS in a real-time dynamic network environment.

Keywords: Adaptive Learning, Intrusion Detection System, Snort, Zeek, Zeek-Flowmeter, Metasploit, Scapy, CICIDS, PCAPs

II. Introduction

Traditional intrusion detection systems (IDS) are increasingly challenged in detecting and neutralizing sophisticated, evolving cyber threats in today's rapidly changing cybersecurity landscape. Systems like Zeek, Suricata, and Snort [1] [2] struggle to keep pace with new threats due to the sheer volume and complexity of network data. To overcome these limitations, there is a growing need for adaptive learning methods and advanced Deep Learning (DL) models within current IDS frameworks.

Suricata and Snort 3 are signature-based intrusion detection systems. Compared to Suricata, Snort-3 offers better performance, requiring less computational power and operating more efficiently. However, Snort-3 cannot detect unknown attacks without the integration of Machine Learning (ML) or Deep Learning (DL) models [3]. As the threat landscape constantly evolves, traditional IDS may find it difficult to keep up with modern attack methods. IDS must adapt to effectively identify and address new threats as cyber attackers continue to refine their tactics. This study explores innovative ways to enhance the accuracy and flexibility of current IDS systems to address their shortcomings. Conventional IDS face challenges in reliably identifying complex and dynamic cyber threats. Machine learning, which can adapt to evolving threats and analyze patterns, offers significant potential for improving

intrusion detection accuracy and efficiency. This work aims to apply machine learning techniques to address the limitations of traditional IDS.

The study employs Machine Learning and Deep Learning models, including Long Short-Term Memory (LSTM), a type of Recurrent Neural Network (RNN) capable of capturing long-term dependencies in sequential data. LSTMs can process and analyze sequential data effectively. Another model used in this study is the Decision Tree, a hierarchical decision support model that employs a tree-like structure of decisions and their possible consequences. Additionally, the study uses a Multi-Layer Perceptron (MLP), an artificial neural network with multiple layers of neurons. The neurons in MLPs typically use nonlinear activation functions, enabling the network to learn complex data patterns. In this study, the MLP helps recognize attack patterns, predict attacks, and generate alerts.

III. Literature Survey

A. Zeek Intrusion Detection System

Zeek one of the most used IDS and also a network traffic analyzer. It can inspect both inbound and outbound network traffic and may be used to identify intrusions based on signatures as well as anomalies. Zeek collects and records network data from monitored devices. Inspite of its many advantages, there is still room for improvement. When individual detectors like Zeek are employed for anamoly detection, they are not capable of producing results as good as when a scheme of a combination of hybrid detectors are used. The hybrid system results in an efficient intrusion detection system that can recompense for the lack of efficiency of the individual detectors. This conclusion can also be confirmed in the experiment by Ngyuen in [1] where the authors proposed a combinational A-IDS in solving the Web attack detection problem.

B. Snort Intrusion Detection System

Snort is an open-source network intrusion detection and prevention system (NIDS/NIPS). It is one of the most extensively used IDS/IPS solutions in the world, noted for its excellent bedside performance, flexibility, and broad rule-based detection features. Snort typically includes one or more network-based sensors that monitor and filter all network traffic. When suspect or malicious network traffic is identified, the sensors assist to filter it and provide alarms. [5] IDS are devices designed to detect unauthorised system access or modification. An Intrusion Prevention System (IPS) is designed to prevent attacks as well as detect them. [6] Thus, we can also use Snort as an IPS. The multithreaded nature of Snort 3 is an advancement which sets it apart from other IDSs.

C. Intrusion Detection Systems and Machine Learning

Studies have been conducted worldwide for the improvement of IDSs in the areas of accuracy, performance, etc. With the help of Machine Learning and Deep Learning Models, Intrusion Detection Systems like Zeek and Snort get a massive boost, in terms of less false positive rates, more true positives and better accuracy. The study performed by Roshandel suggests that integrating both detectors with a hybrid approach will reduce overall weaknesses of individual systems and magnify the detection strengths. [4] Thus, this is the major focus of our study wherein we combine ML/DL models with Zeek and perform a comparative study of the collaborative model against the Snort 3 IDS.

IV. Proposed System

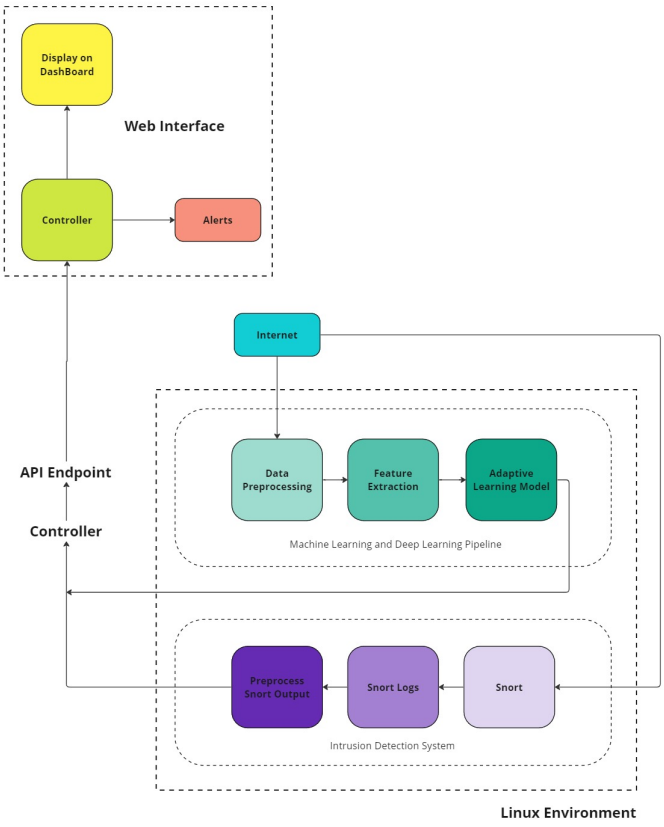

Fig. 1. Functional Block Diagram

Intrusion detection systems(IDS) integrated with machine learning and deep learning models can generate alerts and warnings faster than the standard intrusion detection system. The IDS and ML/DL models require pcap and CSV as input, respectively, to generate the final results. The Zeek IDS converts the pcap file into a CSV which is then fed to the ML/DL models to perform further steps. Metasploit and Scapy are two open source tools that are used to generate pcaps that will be subsequently used to train and test the ML/DL models.

A. Zeek integration with Deep Learning and Machine Learning models

Zeek is a free and open-source software network analysis framework. In this study Zeek will be fed with malicious pcaps and will produce a CSV file as a output. To generate malicious traffic open source tools Metasploit and Scapy are used here.

The generated packets are captured using Wireshark and then stored as a pcap file. The pcap file generated is fed to Zeek which transforms the pcap to Zeek log. We have used Zeek Flowmeter script to convert the pcap to Zeek log. Zeek Flowmeter is a script which extracts the flow data, similar to CICFlowmeter, and create logs. We have used it to generate the flowmeter.log file, which is then parsed using a Python script.

The CSV files derived from the malicious pcaps are now processed, i.e. common attacks are merged, and insignificant attacks are removed from the CSV.The CSV file formed is the final dataset for training and testing. The dataset is normalised using min-max normalisation. This formed data set is used to train and test all the Machine Learning and Deep Learning models such as LSTM, CNN, and MLP. The data is split into train-test data where 70% of the data is used for training and the other 30% for testing. All the models are trained and tested on the generated dataset and the results are compared.

The models trained for this study are Multilayer Perceptron, Decision Tree, Convolution Neural Network and Long Short-term memory.

B. Adaptive learning based IDS

The Adaptive Deep Learning Model has a massive impact in increasing the accuracy of the overall system in comparison to standalone models. This combination of frameworks allows to dynamically determine the best model based on the historical and current performance of the model. Each model is assigned a weight which is variable and updates on the basis of a Mixing Algorithm (Loss and Mixing Updates).

In the current combined framework, we have used four models, namely: Multilayer Perceptron, Decision Tree, Convolution Neural Network and Long Short-term Memory. Initially, all models have an equal weight which gets updated subsequently based on how accurately they predict attacks and benign traffic. Each model's pipeline gives some output for the current traffic which is fed to the ADL algorithm. A model's weightage increases for every correct prediction and is penalized for an inaccurate prediction. The component n, commonly known as the learning rate parameter, is determined in advance. The weight of the i-th IDS with inaccurate prediction is doubled by a factor $e^{-\eta L_{t,i}}$ and then renormalized. If the prediction is wrong at time t for the ith model, then the ith model is penalized at time t+1.

Algorithm 1 Weighted Adaptive Algorithm

Require: res - result, X - Input data matrix, $list\_of\_pipes$ - List of models, N - Number of models, T - Number of flow files, n - Learning rate, a - Weight of historical data
$N \leftarrow$ Length of $list\_of\_pipes$
$T \leftarrow$ Number of rows in X
$x \leftarrow$ None
Initialize v as an array of size $(T+1) \times N$ with all elements initialized to $1/N$
$t \leftarrow 1$
$y \leftarrow 0$
Initialize an empty list res
while $t \leq T$ **do**
  $x \leftarrow$ Prediction of the first model in $list\_of\_pipes$ on X (verbose=False)
  for $i = 1$ **to** N **do**
    Predict using $list\_of\_pipes[i]$ on X (verbose=False) and concatenate the result with x
  end for
  Append a row of zeros to x
  $y \leftarrow$ Array of predictions based on weighted sum of $v[t]$ and $x[t]$
  Calculate loss L as $(y - x[t])^2$
  Update $v[t]$ by multiplying with e^{-nL} and normalize $v[t]$

  if $t + 1 \leq T$ **then**
    Update $v[t+1]$ using weighted combination of $v[t]$ and historical data $v[:t]$
  end if
  Increment t by 1
  Append $y[0]$ to res
end while
return Array res

C. Integration of Adaptive Learning based IDS and Zeek

The integration of the Zeek IDS and the Adaptive Deep Learning model provides us with real time threat alerts. In the previous sections Machine Learning models such as LSTM, CNN, MLP and decision tree were discussed. The mentioned ML/DL models are loaded first and the current attack count for real time traffic is set to zero. Subsequently log files from zeek flowmeter are read i.e. real time traffic is read. The repeated logs generated for the real time traffic are removed from the log file and the new logs are added. This avoids duplication and only distinct logs to be stored in the log file. The unused features are dropped from the log file and flow duration is transformed to match the trained data.

The adaptive learning based ids mentioned in the previous section comes into play. As zeek produces logs for the real time traffic, the pre-trained ml/dl models run in the background to predict on the real time traffic thus increasing the attack counter if a threat is detected and generating alerts for the threat.

D. Snort3 Intrusion Detection System

To perform a comparative analysis of the proposed Adaptive learning based IDS, we have included the Snort3 IDS component in our study. However, during a few recent security assessments, it was found that the default Snort3 community ruleset was inadequate in detecting newly discovered and modified attacks, particularly Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. This inadequacy was tackled by developing and integrating a set of custom rules with reference to the findings of Gupta, Alka and Sharma, Lalitsen. This study put forth a set of rules particularly for the

DoS, DDoS and Port Scan attacks based on the CICIDS 2017 Dataset. Some of the rules are as follows:

TABLE I
SNORT3 UPDATED RULE SET

Attack	Rule
UDP Flood	`alert udp !$HOME_NET any -> $HOME_NET !53 (msg: "UDP FLOOD"; flow: stateless; detection_filter: track by_dst, count 90000, seconds 57; sid:10000008; rev:001;)`
Land Attack	`alert tcp any any -> $HOME_NET any (sameip; msg:"LAND Attack"; sid:10000002; rev:001;)`
ICMP Flood	`alert icmp !$HOME_NET any -> $HOME_NET any (msg:"ICMP FLOOD"; itype:8; detection_filter: track by_dst, count 90000, seconds 69; sid:10000010; rev:001;)`
TCP Reset Attack	`alert tcp any any -> $HOME_NET 80 (flags:R; msg:"Possible DDoS TCP attack"; flow:stateless; sid:10000005; rev:001;)`
Smurf Attack	`alert icmp any any -> 172.168.18.65 any (msg:"DDoS Attack"; itype:0; detection_filter: track by_dst, count 50000, seconds 74; sid:10000012; rev:001;)`

After the updating of the rule set, different types of attacks were simulated using Scapy and Metasploit. Snort3 showed better results and detected all the simulated attacks. As far as these results were accurate, real time traffic is much more unpredictable and attack patterns change everyday. As a consequence, the Snort3 rule set are in a constant need to be updated frequently so as to include and be aware of the new types of attacks.

V. DATA ACQUISITION AND MODEL TRAINING

The data gathering procedure begins with producing both attack and benign traffic using the Metasploit Framework and Scapy to mimic different attack scenarios. Wireshark is then used to collect network packets, resulting in a raw network activity dataset. Zeek is then used to extract logs and characteristics from the network traffic, allowing for a more in-depth study of the acquired data. Following this, the obtained data is preprocessed and formatted to guarantee compliance with the future analysis and modelling procedures. This involves noise reduction, resolving missing values, and formatting the data into an appropriate structure for subsequent processing. Overall, these processes lay a solid basis for gathering and preparing the dataset required for future model training and assessment. Fig.2 shows the step by step procedure for gathering as well as training the models.

The attack creation method uses both the Metasploit framework and the Scapy framework to simulate various attack

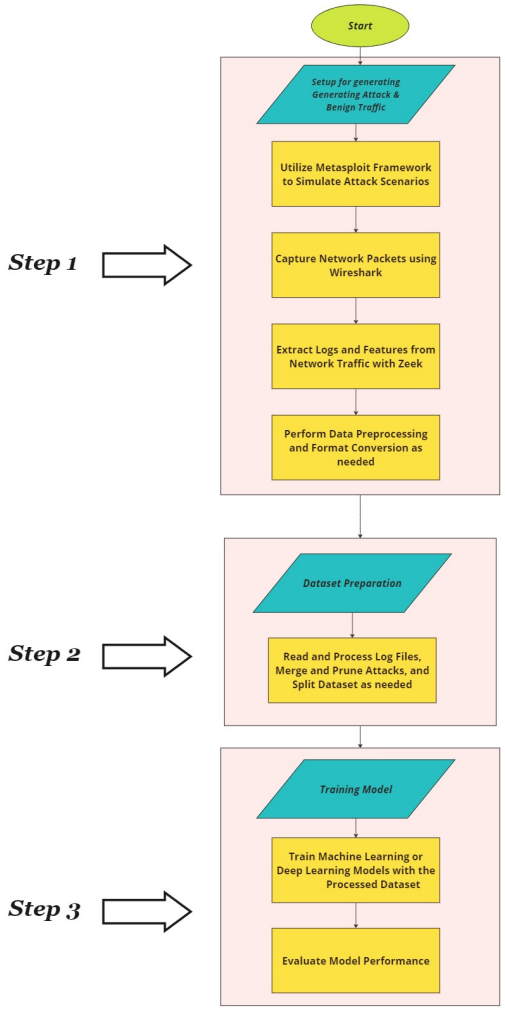

Fig. 2. Sequential stages in setting up the environment, acquiring and preprocessing data, training machine learning and deep learning models, evaluating model performance, and preparing the dataset for training and testing.

scenarios. Metasploit includes a complete set of tools and exploits for initiating attacks on target computers. Metasploit's extensive library of known vulnerabilities and attack tactics allows it to generate complex assaults that are customised to specific targets. Scapy is also a strong packet manipulation tool, allowing users to create packets to launch attacks or exploit weaknesses.

Table II shows the Scapy code for UDP Flood and Land Attack.

TABLE II
SCAPY CODE FOR ATTACKS

Attack	Code
UDP Flood	send (IP(dst = "0.0.0.0", src = RandIP()) / UDP (dport = RandShort()), count = 2000)
Land Attack	send (IP (dst = "0.0.0.0", src = RandIP()) / TCP (dport = RandShort()), count = 2000)

Table III shows the command line code for the attacks. This includes nmap tool for NMAP Decoy Attack.

TABLE III
COMMAND LINE CODE FOR ATTACKS

Attack	Command
NMAP Decoy	nmap -D RND:120 0.0.0.0

Table IV shows the Metasploit code for the attacks. This includes Metasploit tool SYN Flooding and Port Scan.

TABLE IV
METASPLOIT COMMAND FOR ATTACKS

Attack	Command
SYN Flooding	$msf6 > useauxiliary/dos/tcp/synflood$ $msf6auxiliary(dos/tcp/synflood) > setINTERFACEwlo1$ $msf6auxiliary(dos/tcp/synflood) > setNUM2000$ $msf6auxiliary(dos/tcp/synflood) > setRHOSTS0.0.0.0$ $msf6auxiliary(dos/tcp/synflood) > exploit$
Port Scan	$msf6 > usescanner/portscan/syn$ $msf6 > setRHOSTS192.168.1.0/24$ $msf6 > setverbose1\ msf6 > run$

Together, these frameworks create a flexible platform for creating a wide range of attack traffic, from basic vulnerabilities to more complex attack vectors. The attack generation method tries to provide a realistic and representative dataset for further analysis and model training in intrusion detection systems by carefully configuring and executing.

VI. EVALUATION PARAMETERS

A. Recall

Recall in machine learning refers to the proportion of relevant instances (true positives) that were correctly identified by the model out of all the actual relevant instances (true positives and false negatives).it indicates the model's ability to correctly detect positive instances from the entire pool of actual positives. High recall means the model can effectively capture most relevant instances but may also include false positives.

B. Precision

The Precision parameter is the proportion of true positive instances among all the instances that the model predicted as positive (true positives and false positives). It reflects the accuracy of the model's positive predictions. High precision indicates that the model tends to make fewer false positive predictions, thus making it more reliable in identifying positive instances.

C. F-Score

F-score, also known as F1-score, is a statistical and machine learning metric used to assess the accuracy of a binary classification model. It combines accuracy and recall to get a single score that balances both parameters. The F-score goes from 0 to 1, with 1 representing perfect precision and recall and 0 indicating no correct predictions. It is widely employed in situations when you want to strike a compromise between precision and memory, such as information retrieval or medical diagnosis.

D. Loss

In the context of deep learning and machine learning, "loss" is a measure of how closely a model's predictions match the actual target values (also known as ground truth) while training. The loss function, also known as the cost function, quantifies the disparity and offers input to the model, allowing it to alter its parameters to enhance performance.

E. Accuracy

In machine learning, accuracy is a typical assessment parameter used to assess a model's performance. It denotes the percentage of correctly categorised occurrences out of all examples analysed.

VII. EXPERIMENTAL SETUP

Fig. 3 shows the experimental setup for Intrusion Detection Systems through Adaptive Learning and Deep Learning Integration. We employed system specifications which included an Ubuntu 22.04 (64-bit) operating system having 16 GB of RAM, and 16 MB of video memory. In our experimental setup, we added Deep Learning and Machine Learning models to the given system to improve network security. Initially, live traffic goes via Zeek, which transforms packet data to a preset CSV format. This CSV data is then put into a Python script, which runs in the background. The script uses Deep Learning and Machine Learning models to analyse data and predict the existence of malicious packets. Based on these predictions, alerts are generated to warn users about potential security issues.

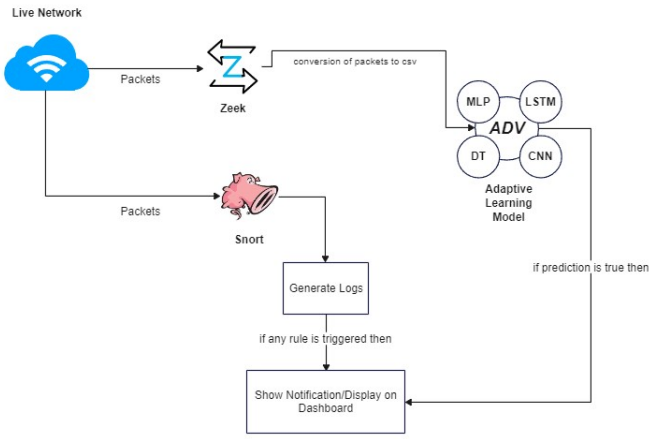

Fig. 3. Experimental Setup for IDS using Adaptive Learning and Deep Learning Integration

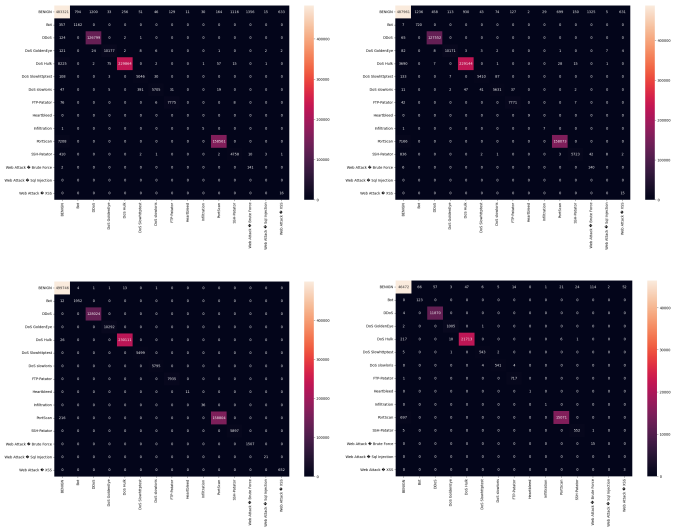

Fig. 4. (a) CNN (b) MLP (c) LSTM (d) DT for CICIDS2017 Dataset

VIII. RESULTS

In this section, we embark on a comprehensive process involving the training and evaluation of our model across various stages. Initially, we undertake the training and evaluation phases using the CICIDS2017 dataset. Following this, we repeat the process by training and evaluating the model on generated data. Finally, we shift our focus to real-world settings, where we subject the model trained on the generated dataset to real world testing to assess its effectiveness.

A. Models based on CICIDS2017 Dataset

The CICIDS2017 dataset comprises both benign traffic and current common assaults, resembling real-world PCAP data. It also encompasses outcomes of network traffic analysis conducted through CICFlowMeter, which categorizes flows based on timestamps, source/destination IP addresses, ports, protocols, and attack types (in CSV files). Data collection spanned Monday to Friday, encompassing five days. Mondays typically exhibited lower traffic levels. The array of attacks includes Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS, occurring across mornings and afternoons on Tuesdays through Fridays.

TABLE V
MODEL TESTING ACCURACY ON CICIDS2017 DATA

Model	Accuracy	Precision	Recall	F-score
CNN	0.9849	0.9858	0.9849	0.9852
MLP	0.9780	0.9804	0.9780	0.9789
LSTM	0.9827	0.9853	0.9827	0.9837
DT	0.9997	0.9997	0.9997	0.9997
AL	0.9862	0.9881	0.9863	0.9870

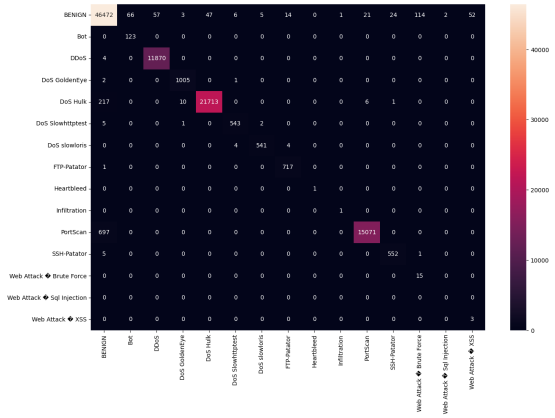

Fig. 5. Adaptive Learning confusion matrix for CICIDS2017 Dataset

B. Models based on Generated Dataset

The dataset contains flow data of malicious PCAPs generated using Metasploit and captured by Zeek-flowmeter. It includes 41,004 UDP flood, 36,907 land attack, 24,171 TCP SYN flood, 20,000 flood DoS (HTTP and ICMP floods), 6,081 decoy attack, 1,652 port scan attack, and 554 benign traffic records. Training accuracy and loss for CNN, MLP, LSTM, Decision Tree (DT), and Adaptive Learning (AL) models are shown in the table VI

TABLE VI
MODEL TESTING ACCURACY ON GENERATED DATA

Model	Accuracy	Precision	Recall	F-score
CNN	0.9909	0.9908	0.9904	0.9903
MLP	0.9847	0.9859	0.9847	0.9845
LSTM	0.9930	0.9929	0.9927	0.9926
DT	0.9998	0.9998	0.9998	0.9998
AL	0.9928	0.9930	0.9928	0.9927

The confusion matrix for each model is shown in Fig 6. This confusion matrix summarizes the performance of the model on the complete data including the training and the testing data.

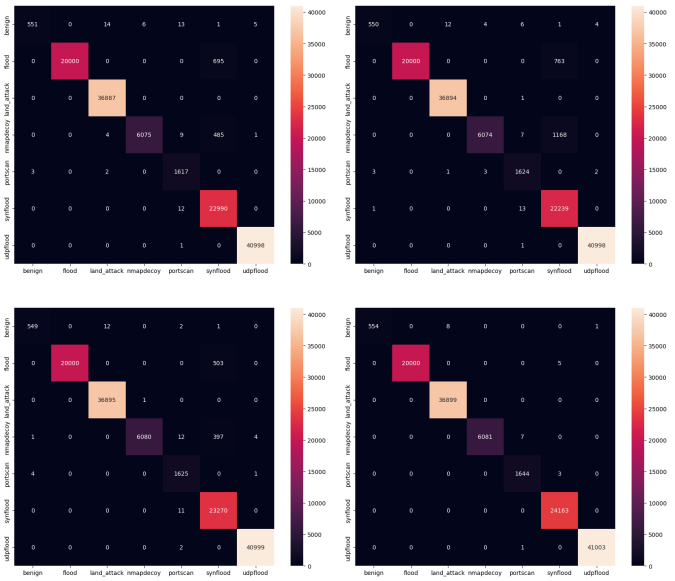

Fig. 6. (a) CNN (b) MLP (c) LSTM (d) DT for Generated Dataset

Here we can observe that the false positives are much less compared to the models trained on CICIDS2017 Dataset.

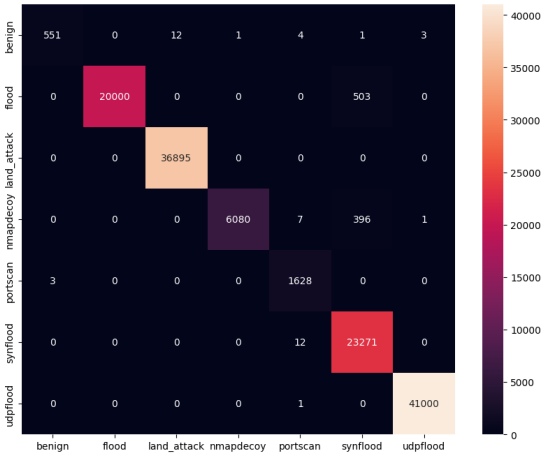

Fig. 7. Adaptive Learning confusion matrix for Generated Dataset

The above diagram shows the confusion matrix for the adaptive learning model on the complete dataset.

C. Testing on real-time traffic

We conducted tests on models trained with the generated dataset. To facilitate real-time monitoring and analysis, we developed a dashboard that visualizes all identified assaults using graphs, bar charts, and detailed attack logs. This dashboard also integrates Snort detection logs, providing a thorough overview of network security events. It is essential to our research, enabling us to track and display the system's performance in real time.

Figure 8 shows the total count of suspicious packets and total packets received. It also shows the distribution of different attacks. Here we can see 2000 Land Attacks where detected and 1 syn flood packet received.

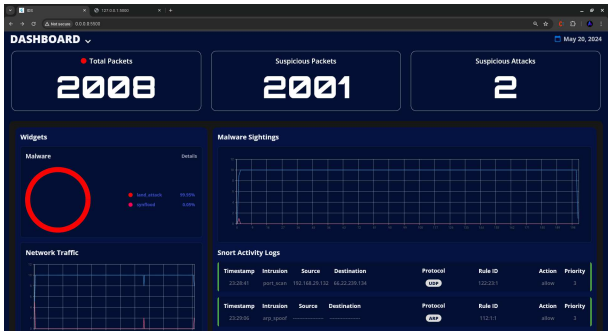

Fig. 8. Dashboard snapshot on live traffic

Table VIII gives the summary of performance of the Adaptive Learning IDS on real-time traffic. These metrics were calculated from the dashboard statistics, by performing actual attacks on the host machine using ScaPy, Metasploit and command line.

TABLE VII
MODEL TESTING ACCURACY ON TEST DATA

Attack	Accuracy	Precision	Recall	F-score
Port Scan	0.9295	1.0000	0.8740	0.9328
Syn flood Attack	0.9438	0.4537	1.0000	0.6242
UDP Flood attack	0.9995	1.0000	0.9898	0.9949
Land Attack	0.9995	1.0000	0.8750	0.9333
NMap Decoy	0.8083	1.0000	0.7125	0.8321

An analysis for CPU, Memory and Virtual Memory consumption is performed for the system. The Results were obtained using the top command in linux. This data is then analysed in python and a plot is created using matplotlib and pandas library.

TABLE VIII
PERFORMANCE OF IDS

Resource Utilization	Mean Performance
CPU	8.936634%
Memory	8.250495%
Virtual Memory	4.8228 MB

A UDP flood and a TCP flood is performed in successions during the duration of collecting the resource utilization of resources. The attack's effects are reflected on the performance graph in Fig. 9 and Fig. 10. It can be seen in the figures that the total resource consumed is directly related to the number of attacks detected.

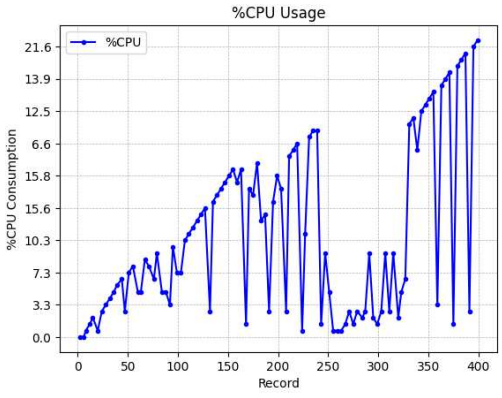

Fig. 9. CPU Usage

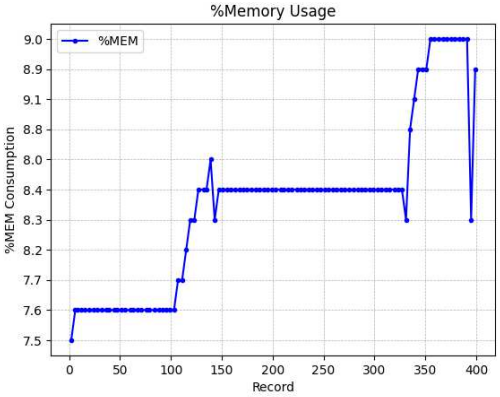

Fig. 10. Memory Usage

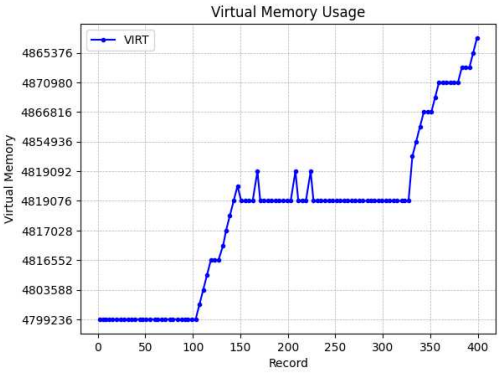

Fig. 11. Virtual Memory Usage

IX. CONCLUSION

The Adaptive Intrusion Detection System (IDS) showed strong performance when trained on the CICIDS2017 dataset to test its architecture, achieving 0.9862 accuracy, 0.9881 precision, and 0.9862 recall, indicating low false positive and false negative rates. For actual implementation with Zeek-Flowmeter, the model was trained on the generated data, achieving accuracy to 0.9928 with high precision and recall due to an optimized environment and superior log extraction. In live traffic tests, the model detected various attacks effectively with an average accuracy of 0.9361 and efficient resource usage, peaking at 22% CPU and 9% memory. These results confirm that the Adaptive IDS performs well in real-world scenarios with good accuracy and resource efficiency.

REFERENCES

[1] Ouiazzane, Said, Malika Addou, and Fatimazahra Barramou. "A Suricata and Machine Learning Based Hybrid Network Intrusion Detection System." Advances in Information, Communication and Cybersecurity: Proceedings of ICI2C'21. Springer International Publishing, 2022.

[2] Fekolkin, Roman. "Intrusion detection and prevention system: overview of snort and suricata." Internet Security, A7011N, Lulea University of Technology (2015): 1-4.

[3] Vadhil, Fatimetou Abdou, Mohamedade Farouk Nanne, and Mohamed Lemine Salihi. "Importance of machine learning techniques to improve the open source intrusion detection systems." Indonesian Journal of Electrical Engineering and Informatics (IJEEI) 9.3 (2021): 774-783.

[4] Dede Fadhilah and Marza Ihsan Marzuki, "Performance Analysis of IDS Snort and IDS Suricata with Many-Core Processor in Virtual Machines Against DOS/DDOS Attacks" Electrical Engineering Department Universitas Mercu Buana Jakarta, Indonesia

[5] Alka Gupta1* and Lalit Sen Sharma, "Mitigation of DoS and Port Scan Attacks Using Snort", Department of Computer Science and IT, University of Jammu, Jammu, India.

[6] S Chakrabarti, M Chakraborty and I Mukhopadhyay, "Study of Snort-Based IDS", International Conference and Workshop on Emerging Trends in Technology (ICWET 2010) – TCET, Mumbai, India

[7] Somayeh Roshandel, "Performance Analysis of a Graph-based Anomaly Detector and the Zeek Intrusion Detection System" University of Victoria 2021.

[8] Tianshuai Guan, "MACHINE LEARNING BASED IDS LOG ANALYSIS" Department of Computer and Information Technology West Lafayette, Indiana May 2021

[9] IPS gaining ground over IDS. (2005, February 14). Network World

[10] Kim, Jiyeon, et al. "CNN-based network intrusion detection against denial-of-service attacks." Electronics 9.6 (2020): 916.

[11] Laghrissi, F., Douzi, S., Douzi, K. et al. Intrusion detection systems using long short-term memory (LSTM). J Big Data 8, 65 (2021).

[12] Carneiro, José, et al. "Machine learning for network-based intrusion detection systems: an analysis of the CIDDS-001 dataset." International Symposium on Distributed Computing and Artificial Intelligence. Cham: Springer International Publishing, 2021.

[13] A. H. Lashkari, M. S. I. Mamun, G. Draper Gil, and A. A. Ghorbani, "Characterization of TorTraffic using Time based Features," 2017.

[14] I. Sharafaldin, A. Lashkari Habibi, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," 2017, pp. 108–116.

[15] Hai Thanh Nguyen and Katrin Franke, "Adaptive Intrusion Detection System via Online Learning" Norwegian Information Security Laboratory Gjøvik University College, Norway.

[16] Jiyeon Kim, Jiwon Kim, Hyunjung Kim, Minsun Shim and Eunjung Choi "CNN-Based Network Intrusion Detection against Denial-of-Service Attacks", Center for Software Educational Innovation, Seoul Women's University, Seoul 01797, Korea;

[17] Hänninen, Markku. "Open source intrusion detection systems evaluation for small and medium-sized enterprise environments." (2019).