INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

OPEN ACCESS

Received: 30-11-2023

Accepted: 23-03-2024

Published: 15-04-2024

Citation: Abirami A, Lakshmanaprakash S, Priya RL, Hirlekar V, Dalal B (2024) Proactive Analysis and Detection of Cyber-attacks using Deep Learning Techniques. Indian Journal of Science and Technology 17(15): 1596-1605. https://doi.org/10.17485/IJST/v17i15.3044

* **Corresponding author**.

abi.lecturer@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Proactive Analysis and Detection of Cyber-attacks using Deep Learning Techniques

A Abirami[1]*, **S Lakshmanaprakash**[1], **R L Priya**[2], **Vaishali Hirlekar**[3], **Bhargavi Dalal**[4]

1 Department of Information Technology, Bannari Amman Institute of Technology, Sathyamangalam, Erode, Tamil Nadu, India
2 Computer Science Engineering, Vivekanand Education Society's Institute of Technology, Mumbai, Maharashtra, India
3 Computer Science Engineering, Shah & Anchor Kitchhi Engineering College, Mumbai, Maharashtra, India
4 Computer Science and Engineering, Nutan College of Engineering & Research, Pune, Maharashtra, India

Abstract

Objectives: This study objective is to create a proactive forensic framework with a classification model to identify the malicious content to avoid cyber-attacks. **Methods**: In this proposed work, a novel framework is introduced to analyze and detect network attacks before it happens. It monitors the network packet flow, captures the packets, analyzes the packet flow proactively, and detects cyber-attacks using different machine learning algorithms and Deep Convolution Neural network (CNN) technique. The KDD dataset is used in this experiment with 30% for testing and 80% for training. **Findings**: The simulation results show that the detection percentage of the proposed framework reaches a maximum of 95.92% in different scenarios. It is approximately 10% higher than the existing proactive frameworks for example Gawand's model, Ahmetoglu's model and many more. **Novelty and applications**: The proposed framework is a proactive model which detects the cyber-attack in prior to avoid cyber-attacks. The deep CNN model highly efficient for detecting cyber-attack.

Keywords: Proactive Forensic Framework; Deep CNN; Classification Algorithms; Cyber attack detection; Intrusion Detection System

1 Introduction

The goal of network forensics is to figure out how security was violated and take preventative actions in the future[1]. The cyber-crime network investigation framework can be categorized into two types such as reactive and proactive investigation.

Reactive network forensic investigation or a post-mortem approach is done only after cyber-crime. It identifies the cause of cyber-attack, preserves the remaining data, collects the details of leaked data, and analyzes the environment[2]. It collects both active and reactive evidence of the attacks. After an occurrence, active evidence refers to gathering all live (dynamic) evidence that exists. Processes running in memory are an example of such evidence, whereas reactive evidence pertains to gathering all the static evidence that remains, such as a hard disc image. However, both active and reactive evidence collected from cyber-crime may remain incomplete. Reactive approaches make it harder to prove in court that the available data was used for data collection, legitimate proof against illicit conduct, or network intrusion detection[3]. Reactive methods are time-consuming, costly, and have a high possibility of errors and requires more effort to analyze a large amount of evidence.

The second type of digital forensic is a proactive method that intends to identify the cyber-attacks before it affects the system. Proactive forensic is an early warning system that uses high-level futuristic rules and machine learning algorithms to monitor the live packet flow and identify the unusual behaviors in the network. Also, it detects anomalies in network traffic and unauthorized alterations of system configurations.

For the IoT environment, Islam proposed a comprehensive Digital Forensic Investigation architecture that allows for more efficient and effective investigation and trace gathering. For digital forensic specialists and experts, it gave a more understandable DFI framework. It lowers reliance on the Cloud Service Provider (CSP) while the investigation is underway[4]. It analyses the frequency of threat occurrences using the Elcat algorithm. By collecting network packets and scanning the data for harmful material, Makwana et al.[5] investigated several scenarios. The experiment is done only with the restricted protocols by scanning the ports. The experiment is still conducted using the network-collected log, which once more serves as a reactive forensic model.

A digital forensics system designed by Dimitriadis et al.[6] for analyzing and examining cyber-attacks, focuses on improving the inspection and analysis stages. The framework first suggests categorizing digital artifacts and linking them with the Cyber-Kill-Chain attack phases. Second, it offers thorough instructions for the phases of analysis and examination. An application with a typical spear phishing attempt is used to demonstrate the usefulness of D4I. This different phased model is defined in 2020. The cyber-attack analysis paradigm for the planned test platform for cyber-attacks and defenses is analyzed by Qi et al.[7] in the year 2020. The framework uses the preliminary findings from detection as its input to analyze the current cyber state using the Cyber security Knowledge Graph (CSKG) and association analysis approach. The CSKG module along with the association analysis module make up the framework's foundation.

Qureshi et al.[8] used various network infrastructures to analyze various inquiry models. Using various network scenarios, evaluation was carried out for both internal and external cyber attacks. The investigation is conducted using the network logs, which lead to the conclusion that the reactive forensic model was used for testing. Firdonsyah et al.[9] reviewed many investigative models and recommended a framework that would work for various organizations. The investigation models reviewed are the reactive model and described the flaw with the reactive investigation models. A prediction model is required for proactive research in order to detect attacks early and prevent them from being started.

Machaka et al.[10] drafted a proactive forensic architecture in 2022 with 5 phases which suggests that the network to be continuously monitored. The network monitoring server functions as an agent and maintains continual communication with host computers and network components while looking for any unusual system synchronization. In this model, the author used a server for the entire monitoring and there is no prediction model used to identify the malicious content that enters the network. As the server perform the evaluation based on the behaviors of the network, the investigation will be performed which again resemble the reactive forensic framework.

Some of the author's done research on cyber-attack prediction based on machine learning techniques. The author Palanikumar[11] used machine learning algorithms like Support Vector Machine (SVM) and Naïve Bayesian to train the model and validate it using incoming live network packets for Denial of Service (DoS) type of attack; the experimental results show that Naïve Bayesian yields better accuracy of 88% when compared to the SVM model. Ahmetoglu et al.[12] listed each machine learning techniques, data sets, and emphasis on cyber-security. Different classification techniques utilized in these research were compared to find the better one by evaluating the performance using various parameter. The challenges in using machine learning based methods are discussed to improvise the performance.

Gawand et al.[13] applied machine learning techniques to analyze datasets, which determines if given data is normal or abnormal. Algorithms like K Nearest Neighbor, Logistic Regression (LR), Decision Tree, Gaussian Naive Bayes, Support Vector Machine, Gradient Boosting Classifier (GB), and Random Forest (RF) are used for classification, detection, and prediction. The author conclude from the analysis of the data that the system, using a Decision Tree, Random Forest, and Xgboost algorithm, produced improved Accuracy, Precision, Recall, and F1 Score. Gradient Boosting, K Nearest Neighbor, likewise attains a greater accuracy of 94%.

The problem with the reactive forensic framework is that, the cyber-attack launched by the attacker will damage the network, website and many more which depends on the attack and the investigation is done only after the attack. The data related to the investigation can also be destroyed by the attacker. The problem of the reactive model is eliminated by the introduction of the deep learning based classifier in the proposed proactive investigation framework.

In this paper, a proactive forensic investigation model is proposed which proactively monitors, captures and analyzes the network packet flow and effectively detects the cyber-attacks that can occur in a network in the future. The contribution of the proposed work are summarized as follows, It uses Deep Convolution Neural Network (CNN) technique to analyze the packet flow in the network system. The proposed framework is modeled with the option for generating network traffic, inducing the flow of data in the network. The live packets arriving at the system are captured. The proposed CNN-based deep learning technique effectively analyze the incoming packets and group them under different categories. The packet flow is analyzed for a particular period and deviation is predicted from the incoming packets. Also, the proposed framework can generate network packets, frame an attack, and analyze the packet flow before and after the attack and earlier detection of attacks based on packet flow. Since the proposed framework is modeled with proactive forensic, it enables ease of analysis and detection of cyber-attacks before it happens. On the proposed framework, several numerical studies are conducted about flow allocation, cyber-attack latency, cyber-attack detection accuracy, and evaluating the effects of the disaster and response on a cyber-physical system.

The paper is organised in such a way that Section 2 highlights related research on reactive and proactive digital forensic methodologies. The numerous types of attacks are discussed in Section 3. The proposed intelligent framework for proactive investigation is discussed in Section 4. Section 5, discusses the machine learning algorithms used in the framework to detect cyber-crimes. Section 6 presents the simulation results, and Section 7 brings the work to a close.

2 Methodology

Existing reactive digital forensic methods are time-consuming and cost-ineffective methods. They lag in with proper packet classification methods and security implementations. Existing reactive digital forensic methods requires a lot of attack signatures and a list of packet behaviours to create an efficient intrusion detection statutes. The proposed Deep Convolutional Neural Network (CNN) based method frames their own rules with respect to the empirical data and attack histories. Also, the proposed framework performs live and continuous monitoring on the network. It efficiently sniffs around the network and identifies the area where packet deviation occurs and immediately alerts the concerned authorities whereby taking necessary actions before the attack happens. The suggested system is viewed as an infrastructure that notifies network users when odd behaviour occurs. It is described as a security system that is based on alerts. The suggested system identifies or detects malicious packets hitting the network.

The proposed intrusion detection system has the following basic functions: (i) Reading the incoming packets using Wireshark live packet capturing tools, (ii) Classifying the packet, (iii) Training the intrusion detection model using deep CNN algorithm[(14)] and KDD dataset[(15)], (iv) Finding malicious packets in the network, (v) Raising alarm to the authorities and the authenticated users, (vi) Protect the system from the predicted cyber-attacks, and (vii) Continuously monitoring the network.

A private network was created with 40 computer systems, and video files were transferred. To sniff the packet that are transferred inside the network, Wireshark network analyser software is used. Once after the packets are captured, it is stored in the packet storage of the cyber physical system (CPS). It is an intelligent system which executes the CNN based deep learning technique to classify the live packets.

Later, the packets are sent to the classifier module, where the packets are classified into three types such as: (i) Flow change packets, (ii) Time change packets, and (iii) Sign change buckets. The Recursive Flow Classification (RFC) heuristics algorithm[(16)] is used to classify the packets in the proposed framework.

Recursive Flow Classification (RFC) is a heuristic algorithm, which is used for live packet classification on the proposed framework. The RFC algorithm[(17)] classifies packets by mapping the packet header bits 'S' to a bit action identifier 'T,' wherein T = log N, T<<S. RFC constantly calculates the action for every one of the 2^S dissimilar data packet headers. RFC tries to do any such mapping multiple times in order to get the best result. At each level, the algorithm converts one set of values into a smaller set. During each phase, a sequence of memories returns a value that is less than the memory access index (i.e., represented in less bits). Algorithm 1 explains the processing stage of the RFC heuristics method[(18)] and Figure 1 explains the workflow of the same[(19)].

Algorithm 1- RFC Heuristics packet classification method

Input: Live Packet

Output: Classified Packets

Step 1: Fields in the packet header are divided up into various parts during the first stage, which are then utilised to index into several memories simultaneously. Each memory's contents are chosen in such a way that the lookup outcome is smaller than that of the index.

Step 2: Memories are catalogued in succeeding phases based on the outcomes of previous phases.

Step 3: The memory gives way to action in the last step.

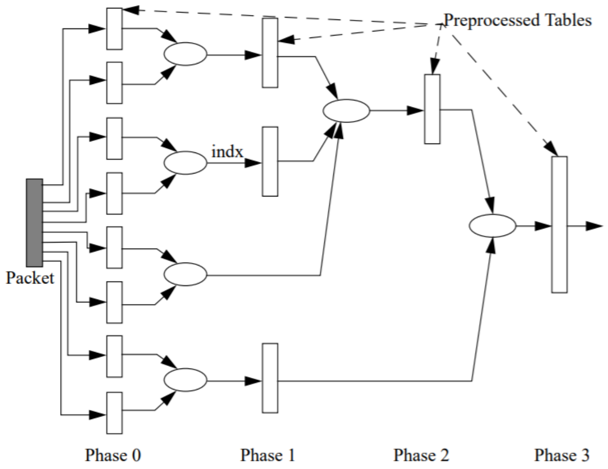

Fig 1. Data Packet classification using RFC heuristic method

Algorithm 2 - Tracing abnormal behavior

```
Input: The captured packets P1, P2, P3,.....Pn
Output: Tracing abnormal behavior
Step 1: Set threshold points ={tp1,tp2,tp3,...tpn}
Step 2: while(1)
if the stopping criterion is satisfied then
  return result set
  else
Step 3: while(packet flow in the system)
  Choose the feature set Fi
Filter the packets based on the threshold points
Trace the feature set in the packet
  for each value Vj of attribute Pi do
  create datasets d[Vj] based on Pi
  recursively build a subtree by using a corresponding subset of datasets di
  end for
end while
return result set
end if
where the stopping criterion is reached
Step 4: End
```

Let n$_i$, n$_2$, ...n$_n$ be the set of internal nodes and s$_1$,s$_2$,...s$_n$ be the set of external nodes involved. Let l$_1$, l$_2$,...l$_n$ be the links established among the nodes. Let r$_1$, r$_2$, ... r$_n$ be the rate of information flow between nodes. The overall transmission of normal traffic flow be λ_i = s$_i$+$\sum$r$_i$n$_{i.}$ (20).

Let n$_i$$\in$n be the compromised node. Let P$_s$ denotes the packet flow per second. F$_s$ denotes the packet flow in both direction and M$_s$ denotes the maximum transmission unit size Let the flow rate be r(P) = {Ps * Fs * 8 * Ms}. The initial packet flow rate is 3000 packets per second which provides an effective throughput rate of 0.072864Gb/s. This rate is observed for a time flow of t$_i$. Now the traffic flow be deviated as

$$\lambda i \;=\; (si + \sum rini \;\pm\; \{r(P) \;*\; ti\} \tag{1.1}$$

Now through the compromised node ni, the packet flow rate is decreased and increased periodically leading to a throughput lack of ± 2.3276 Gb/s. Let the flow rate be equated to overall delay by TD = L/R where L is the packet length in bits and R is the transmission speed. This misflow details are analysed and reported to the alarmer module. Let c(pi) be the capture of packets by the system for a particular time period ti. The received packets are properly analysed by the algorithm used in the system. The total flow within the system can be calculated using t(P) directly proportional f(P)*nt(P). The overall packets captured by the sytemis f(t[p]) = {r(p)*t(p)}. The received packets are captured and send to the system database using packet capture module. The packets are formatted and filtered to a form that eases down the analysis process.

To train the CNN model in the proposed framework, KDD CUP 1999 dataset is used[21]. The dataset has four different types of attacks such as, Denial of Service (DoS), User to Root (U2R), remote to local (R2L) and probe. The classified packets are further given as an input to the flow module of the proposed Deep CNN model to predict the intrusion in the network. In the flow analyzer module, the packets are grouped into different categories based on the flow, time and format. After grouping the classified package in the flow analyzer model, Deep Convolution neural network is used to detect the anomalies in the network. The alarming system sends an alarm to the respective authorities to alert abnormality in and around packet flow. Finally, a report within detail analysis of the packet flow within the system is sent to the authorities, thereby paving the way for proper action before the crime happens. Figure 2, shows the work flow of the proposed framework. The framework provides a better User Interface where the end user can track the incoming packets in visual format.

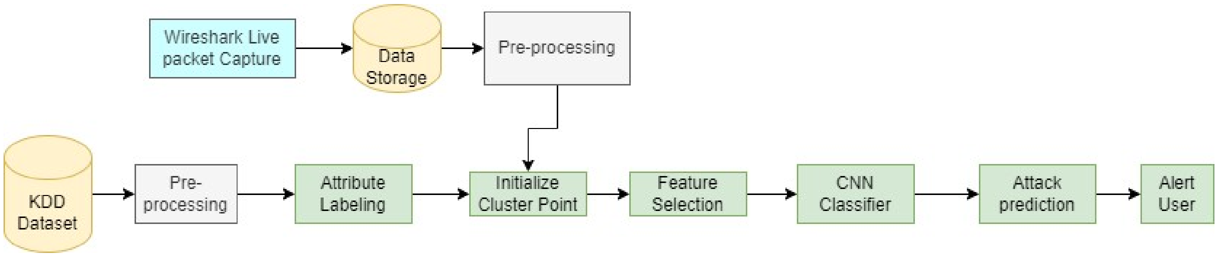

Fig 2. Workflow of the proposed framework

2.1 Data Preprocessing

Because the values in the raw dataset have such a large impact on deciding overall performance, they must be preprocessed. The KDD Cup 98 dataset is first preprocessed to allow for data manipulation using the concepts of normalisation, balancing, segmentation, and elimination of extraneous data. This work employs the Markov Chain Clustering (MCC) methodology, which accomplishes data preprocessing and classification in enhancing data quality. Following the loading of the dataset, attribute labelling, cluster index initialization, and segmentation for arranging feature values can be done. Algorithm 2 demonstrates how to use Markow Chain Clustering to preprocess the KDD Cup 98 dataset.

```
Algorithm – Markov Chain Clustering
Input: Input Data (networking packet details)
Output: Preprocessed and Clustered networking data $TD$.
Step 1: The feature value of the input network data should be normalised.
Step 2: The features in the normalised data are placed in a logical order.
Step 3: Calculate the separation between particles.
Step 4: Compute the Markov model’s weighted estimated value using,
For $i$=1 $to$ $N$
If $((i)<\omega(i+1))$, then
$Ni= \{k, (1+(i))<00, otherwise$
$k = k + 1$
End if
End loop
End
```

2.2 Training the Deep CNN model using the preprocessed

The KDD Cup 98 dataset contains a total of 41 various traffic attributes, 38 of which have been numerical attributes while 3 of which are symbolic attributes. For ease of processing, the symbolic data was converted to numerical data. The symbolic type has three characteristics: a TCP/IP layer’s protocol type, a target system’s service type, and a flag type that indicates the session’s connection state. ICMP, TCP, and UDP are three different protocols. By one encoding, three protocols are turned into three-dimensional vectors (1,0,0), (0,1,0), and (0,0,1). The dataset’s numerical characteristics in the range of 0 to 255 are translated into a 117-dimensional vector, and pictures with 13* 9 pixels are generated. A value between 0 and 255 should be assigned to each colour channel of the image. These pictures are then fed into CNN model as given in Figure 3.

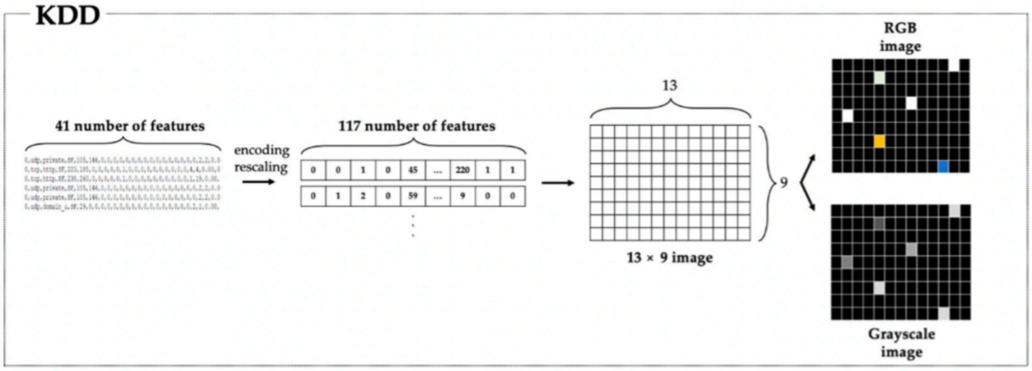

Fig 3. Conversion of numeraical features into a RGB image for CNN training model

2.3 Design of CNN intusion detection model

In the proposed algorithm, the Convolutional Neural Network (CNN) is used to differenate the normal packet flows and the abnormal flows. Figure 4 illustrates the overall design of the proposed CNN intrusion detection model.

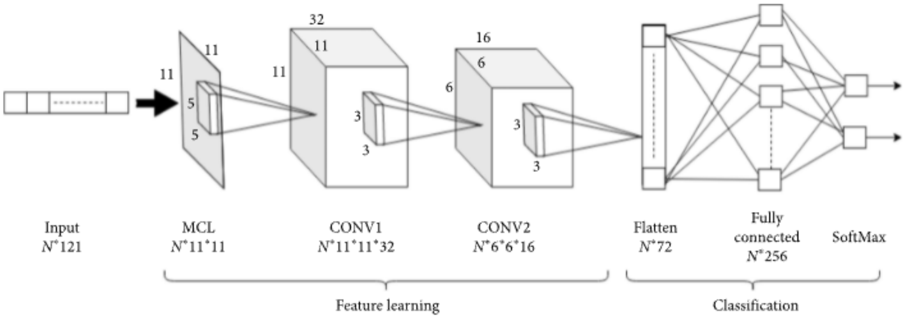

Fig 4. Design of the CNN based intrusion detection model

The most generally used DL model for image identification is CNN, which consists of a convolution layer which collects image information and a fully connected layer that decides which class the input image corresponds to. The convolution layer extracts the image's unique characteristics while preserving the image's I/O and spatial information, and by adding a pooling layer to the convolution layer, the size of the feature data is reduced.

3 Results & discussion

To evaluate the proposed framework, model attacks are created and analyzed the behavior of the system in providing detection against the attack. The KDD dataset is used to train the machine learning based classification algorithms. It assists in the analysis of incoming live packets and the classification of harmful and non-malicious. The proposed framework is trained utilising a variety of machine learning methods, including Decision Trees, Random Forests, Support Vector Machines, and Back Propagation Neural Networks. The accuracy and the precision of each algorithm is compared with one another. The suggested framework was evaluated using software tools such as Eclipse IDE, Weka, and IPmessenger, and the complete project is written in Java.

The analysis of the proposed framework is made on two basis, such as (i) The overall flow allocation and the delay experienced by the system in normal flow and (ii) The overall flow allocation and the delay experienced by the system in malicious flow. The framework is evaluated with different test cases and scenarios.

3.1 Scenario 1: Normal flow analysis

The open source Nping packet generator is used to generate the packets within the network and allows to perform flow analysis and response time measurements. It can generate network packets for a wide range of protocols, allowing users full control over protocol headers. In this testing (normal flow) scenario, twelve wireless nodes were created and three paths are identified from the starting node to destination node. A total of 500 packets each with 128KB is transferred from the start node to the destination node. While the packets are transmitting, the live packets are captured for a limited time using packet sniffing tools and the arrival rate is analyzed. Figure 5 shows the flow allocation and the time taken to transmit the packets through three different paths.

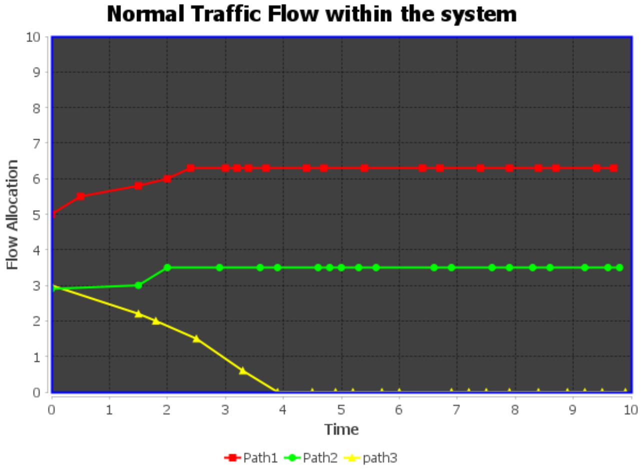

Fig 5. Normal packet flow

3.2 Scenario 2: Attack flow analysis without detection

In this (attack without detection) scenario, the same amount of packets are sent from the starting point to the destination. However, a malicious packet is introduced inside the network and the flow analysis is measured. The values are measured before the attack is detected. Figure 6, shows the malicious packet flow without the detection.

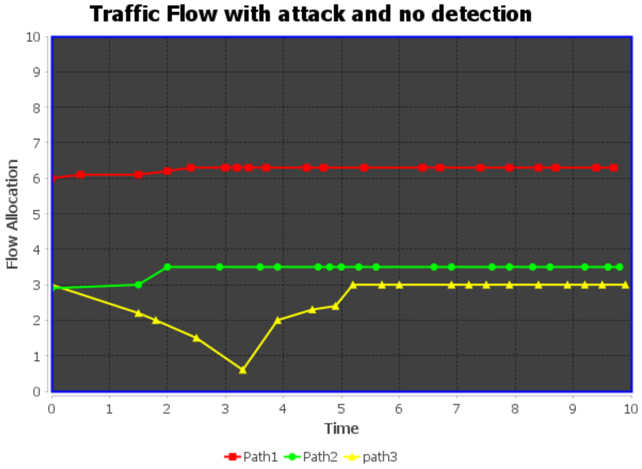

Fig 6. Malicious packet flow without detection

3.3 Scenario 3: Network Attack and detection

In this (attack with detection) scenario, the malicious packets and introduced in the network and flow allocation is measured after the framework detects the malicious packets. Figure 7 shows the traffic flow with malicious attack and detection.

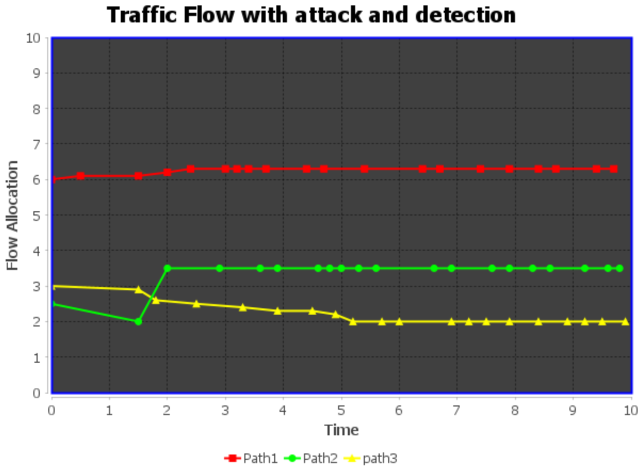

Fig 7. Malicious packet flow with detection

3.4 Comparative Analysis of Machine Learning Algorithms

The proposed system uses different machine learning algorithms to analyze and filter the data packets. The setup of ML algorithm based on Training Model Dataset with the total number of features considered for normal is 20, probe packet is 35, DoS is 25 and R2L is 15 with varying training and testing time.

Various machine learning algorithms are compared with the proposed Deep CNN classification algorithm. Deep learning is a burgeoning branch of machine learning and artificial intelligence research that has been widely and successfully utilised in this fields[22] The algorithm are compared with the parameters like accuracy, Time to build, correctly classified packets and in-correctly classified packets. According to the analysis carried out, the accuracy of classification is better in case of the proposed Deep CNN model, but its takes more time to build the model. Poor accuracy is given by Naïve Bayes Multinomial method, but it takes very less time to build the model. The other classification algorithms are giving an average result.

Table 1. Comparative Analysis of classification algorithms

Algorithms	Accuracy in %	Correctly Classified %	In-correctly Classified %	Time to build in seconds
J48	85.7971	85.7971	14.2029	67.73
Naïve Bayes	92.1398	92.13978	7.860224	5.1
Random Forest	81.7649	81.76487	18.23513	265.1
Naïve Bayes Multinomial	79.2352	79.23521	20.76479	0.09
Bayes Net	86.5957	86.59566	13.40434	20.64
MLP	93.8792	93.87918	6.120817	203.74
BPN	83.7863	83.78625	16.21375	34.82
Deep CNN	95.9263	95.92627	4.07373	109.11

3.5 Analysis of proposed framework across different servers:

The proposed framework is deployed across different servers and their detection rate is analysed for a period of time. Figure 8, shows the performance of proposed framework across different servers. As per the result compiled in the graph, the database server and the resource server are having a constant flow with the detection rate. The web server is with an oscillation and the detection rate is low as compared to the other server. The detection rate is high with the application server.

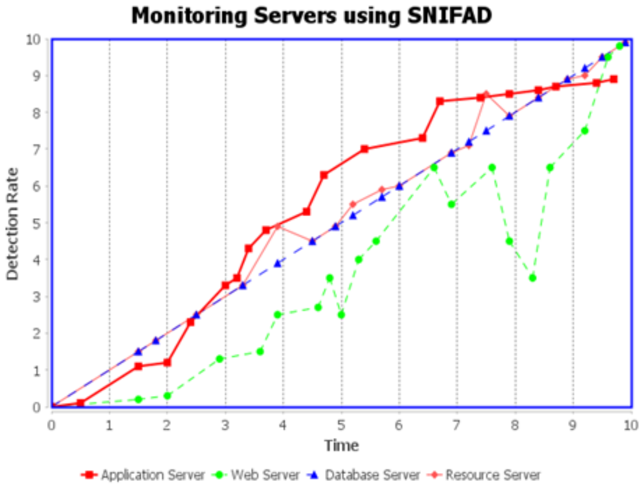

Fig 8. Performance of proposed framework across different servers

3.6 Comparison of proposed framework with existing methods:

The proposed framework is compared with the existing frameworks and the following factors are analysed:

Table 2. The Proposed framework compared with the existing framework

Framework for Digital Forensic Investigation	**Proactive Analysis**								
	Packet Sniffing Preprocessing	**Data Store**	**ML Classifiers**	**Live Alarm**	**Live Report**	**Logger**	**Advance Indexing**	**Registry**	**Live Analysis**
Proposed System	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
DFIDM	Yes	Yes	Yes	No	Yes	No	No	No	No
DFI for IoT	Yes	Yes	Yes	Yes	No	Yes	Yes	No	No
DFI Procedure Model	Yes	Yes	No	Yes	Yes	No	No	No	No
Traceability in DFI	Yes	Yes	No	Yes	Yes	No	No	No	No
Data Analysis of File Forensic Investigation	Yes	Yes	No	Yes	Yes	No	No	No	No

Based on the analysis with the existing framework which is shown in Table 2, it is found that proposed framework outperforms in parameters such as efficiency, speed, live alarming and live report generation. The packets are pre-processed and fed to SQL Server. Using advance indexer the speed of the system is increased compared to the existing works made and eases live alarming and live report generation. Using MQTT background services, efficiency is improved and makes proposed framework the best framework for proactive forensic investigation method.

4 Conclusion

This research work proposes a framework to proactively detect the anomalies and malicious packets by monitoring and analyzing the live packets in the wireless network. The proposed framework works well to capture the overall activity of packet flow, detecting the deviation rate and filter the suspicious activities using deep learning techniques. Threshold values are setup with different parameters taken into account and the steady state of the system is tracked in timely manner. It characterizes the delays experienced by source nodes at the steady- state. By fine-tuning parameters and selecting different algorithms, the attack can be predicted in a proactive way whereby eliminating attacks to be happening saving a lot of time and investigation process. Our framework is set by modelling attacks and proactive investigation in attacks. Also, the framework is modelled to show effectiveness of proactive analyze than reactive analysis. Our simulation results illustrate the trade-off between the different deep learning algorithms with the accuracy of 96% which is better than the existing algorithms. In future work the framework is extended more with user-friendly options for proactive and reactive investigation. The reinforcement learning and quantum computing can be incorporated for the enhancement of the proposed system.

References

1) Abirami S. A Complete Study on the Security Aspects of Wireless Sensor Networks. In: International Conference on Innovative Computing and Communications;vol. 55 of Lecture Notes in Networks and Systems. Singapore. Springer. 2018;p. 223–230. Available from: https://doi.org/10.1007/978-981-13-2324-9_22.
2) Kamil S, Norul HSAS, Firdaus A, Usman OL. The Rise of Ransomware: A Review of Attacks, Detection Techniques, and Future Challenges. In: 2022 International Conference on Business Analytics for Technology and Security (ICBATS). IEEE. 2022. Available from: https://doi.org/10.1109/ICBATS54253.2022.9759000.
3) Abirami A, Palanikumar S. An Artificial Intelligence-based Proactive Network Forensic Framework. *Iraqi Journal of Science*. 2023;64(11):5896–5911. Available from: https://dx.doi.org/10.24996/ijs.2023.64.11.35.
4) Islam MJ, Mahin MD, Khatun A, Debnath BC, Kabir S. Digital Forensic Investigation Framework for Internet of Things (IoT): A Comprehensive Approach. In: 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT). IEEE. 2019. Available from: https://doi.org/10.1109/ICASERT.2019.8934707.
5) Makwana* RRS, and DST. A Network Forensic Framework for Port Scan Attack based on Efficient Packet Capturing. *International Journal of Innovative Technology and Exploring Engineering*. 2019;8(12):4632–4641. Available from: https://dx.doi.org/10.35940/ijitee.l3850.1081219.
6) Dimitriadis A, Ivezic N, Kulvatunyou B, Mavridis I. D4I - Digital forensics framework for reviewing and investigating cyber attacks. *Array*. 2020;5:1–8. Available from: https://dx.doi.org/10.1016/j.array.2019.100015.
7) Qi Y, Jiang R, Jia Y, Li A. Attack Analysis Framework for Cyber-Attack and Defense Test Platform. *Electronics*. 2020;9(9):1–18. Available from: https://dx.doi.org/10.3390/electronics9091413.
8) Qureshi S, Li J, Akhtar F, Tunio S, Khand ZH, Wajahat A. Analysis of Challenges in Modern Network Forensic Framework. *Security and Communication Networks*. 2021;2021:1–13. Available from: https://doi.org/10.1155/2021/8871230.
9) Firdonsyah A, Purwanto P, Riadi I. Framework for Digital Forensic Ethical Violations: A Systematic Literature Review. In: The 8th International Conference on Energy, Environment, Epidemiology and Information System (ICENIS 2023);vol. 448, E3S Web Conf. EDP Sciences. 2023;p. 1–10. Available from: https://doi.org/10.1051/e3sconf/202344801003.
10) Machaka V, Balan T. Investigating Proactive Digital Forensics Leveraging Adversary Emulation. *Applied Sciences*. 2022;12(18):1–15. Available from: https://dx.doi.org/10.3390/app12189077.
11) Palanikumar S, Abirami A. Proactive Network Packet Classification Using Artificial Intelligence. In: Artificial Intelligence for Cyber Security: Methods, Issues and Possible Horizons or Opportunities;vol. 972 of Studies in Computational Intelligence. Springer, Cham. 2021;p. 169–187. Available from: https://doi.org/10.1007/978-3-030-72236-4_7.
12) Ahmetoglu H, Das R. A comprehensive review on detection of cyber-attacks: Data sets, methods, challenges, and future research directions. *Internet of Things*. 2022;20:100615. Available from: https://dx.doi.org/10.1016/j.iot.2022.100615.
13) Gawand SP, Kumar MS. A Comparative Study of Cyber Attack Detection & Prediction Using Machine Learning Algorithms. Research Square Platform LLC. 2023. Available from: https://doi.org/10.21203/rs.3.rs-3238552/v1.
14) Zhang X, Zhang X, Wang W. Convolutional Neural Network. In: Intelligent Information Processing with Matlab. Singapore. Springer. 2023;p. 39–71. Available from: https://doi.org/10.1007/978-981-99-6449-9_2.
15) Wan X, Wang J, Li J, Liu X, Fu X, Wang X. Research on Satellite Quality of Service Classifier Based on Modified Recursive Flow Classification Algorithm. In: 2022 5th International Conference on Information Communication and Signal Processing (ICICSP). IEEE. 2023. Available from: https://doi.org/10.1109/ICICSP55539.2022.10050654.
16) Cai Y. A Multilayer Iteration-Based Improved Recursive Data Flow Matching Algorithm. In: 2021 International Conference on Information Technology and Big Data Engineering (ITBDE 2021);vol. 2074 of Journal of Physics: Conference Series. IOP Publishing. 2021;p. 1–6. Available from: https://dx.doi.org/10.1088/1742-6596/2074/1/012009.
17) Kang J, Kim T, Park J. FPGA-based Real-time Abnormal Packet Detector for Critical Industrial Network. In: 2019 IEEE Symposium on Computers and Communications (ISCC). IEEE. 2020. Available from: https://doi.org/10.1109/ISCC47284.2019.8969630.
18) Dahlman E, Parkvall S, Sköld J. Overall Transmission Structure. In: 5G NR. Elsevier. 2021;p. 115–145. Available from: https://doi.org/10.1016/B978-0-12-822320-8.00007-6.
19) Abirami A, Palanikumar S. BBBC-DDRL: A hybrid big-bang big-crunch optimization and deliberated deep reinforced learning mechanisms for cyber-attack detection. *Computers and Electrical Engineering*. 2023;109(Part B):108773. Available from: https://dx.doi.org/10.1016/j.compeleceng.2023.108773.
20) Kim J, Kim J, Kim H, Shim M, Choi E. CNN-Based Network Intrusion Detection against Denial-of-Service Attacks. *Electronics*. 2020;9(6):1–21. Available from: https://dx.doi.org/10.3390/electronics9060916.
21) John C, Sahoo J, Madhavan M, Mathew OK. Convolutional Neural Networks: A Promising Deep Learning Architecturefor Biological Sequence Analysis. *Current Bioinformatics*. 2023;18(7):537–558. Available from: https://dx.doi.org/10.2174/1574893618666230320103421.
22) Sujatha R, Chatterjee JM, Jhanjhi NZ, Brohi SN. Performance of deep learning vs machine learning in plant leaf disease detection. *Microprocessors and Microsystems*. 2021;80. Available from: https://doi.org/10.1016/j.micpro.2020.103615.