CIP Report – 1

Team Members:

- Abirami Ramanathan (2023103020)
- Hashim M (2023103017)
- Ragotma Ragavendar (2023103023)
- Hariharan B (2023103704)

1. Abstract

In the recent years, the growing complexity and frequency of cyber threats in cloud environments has been dynamically changing in terms of strategy and pattern, which has posed a significant challenge in traditional security mechanisms. This underscores the need for an automated solution to maintain efficient and effective incident response. The project deals with this urgent issue by designing an AI-driven cyber incident response system, specifically for cloud environments, by employing AI and ML techniques to provide accurate, scalable, and seamless integration with cloud platforms like Google Cloud and Microsoft Azure. The three major sub-systems of this framework include a Network Traffic Classifier, Web Intrusion Detection System, and a Post-Incident Malware Analysis, which are integrated by an automated pipeline. This project makes a significant contribution to the field of AI-powered cybersecurity by showcasing the powerful combination of AI models and cloud infrastructure to fill critical gaps in cyber incident response.

2. Introduction

The rapid evolution of cyber-attacks targeting organisations across various industries has highlighted the crucial need for robust incident response capabilities. According to the UK government's Cybersecurity Breaches Survey in 2024, the ENISA Threat Landscape (ETL) 2025 and IBM's Cost of a Data Breach Report, a significant percentage of businesses and charities have experienced breaches or attacks, with alarmingly low adoption rates of formal incident response capabilities (IR). Organisations with well-tested IR capabilities and high levels of AI and ML integration for threat detection and response demonstrated substantially lower data breach costs, as highlighted by IBM's 2024 Cost of Data Breach report.

3. Problem Statement

Despite the alarmingly increasing occurrence and sophistication of cyber-attacks in cloud environments, several organisations heavily rely on traditional and static security mechanisms with limited automation. The lack of integrated, AI-driven incident response capabilities results in delayed threat detection, prolonged response times, increased data breach costs, and heightened operational risks. Existing security solutions often operate in isolation, failing to provide a unified and scalable approach to incident detection, analysis, and response in modern cloud infrastructures. As a result, there is a pressing demand for organisations to invest in incident response capabilities to protect against data breaches and cyberthreats.

4. Objectives of the Project

- To design a containerised AI-driven cybersecurity system that simplifies deployment and efficiently handles large volumes of cloud traffic while maintaining high detection accuracy.

- To develop an integrated threat detection pipeline combining network traffic classification, web intrusion detection, and post-incident malware analysis.

- To implement real-time detection of cyber-attacks using machine learning and anomaly detection techniques across network, web, and file-based attack vectors.

- To deploy a secure multi-VPC cloud architecture that isolates production, DFIR, and honeypot environments while enabling controlled data sharing through VPC peering.

- To provide centralised monitoring and actionable security insights using ELK-based logging, alerts, and dashboards for effective incident response.

5. Scope of the Project

This project focuses on building a scalable, containerised, and AI-driven cyber incident detection system for cloud environments, by integrating three major sub-systems namely the Network Traffic Classifier, Web Intrusion detection System and a Malware Analysis Application. The system is deployed using a secure multi-VPC architecture comprising of the Production environment, DFIR environment and a Honeypot Environment. It is finally analysed by the ELK-Stack for centralised monitoring.

6. Literature Survey

SNo.	Reference	Data Source	Contribution	Novelty and Approach	Limitations
1.	SCARF: A container-based approach to cloud-scale digital forensic processing, by: Stelly and Roussev	Experimental Data	Containerization is used in digital forensics to show that forensic tools can be run as independent, containerised modules instead of a single heavy system.	-Containers are connected in a processing pipeline.--Each container performs a specific task.	The absence of experiments in actual cloud environments limits the assessment of the scalability and full potential of their work.
2.	Forensics as a Service: Three-Tier Architecture for Cloud Based Forensic Analysis, by: Nanda and Hansen	Cloud Resources	Forensics as a Service with VM snapshots.	Implemented a Forensic as a Service (FaaS) solution, that enables digital forensics to be conducted through a cloud-based Forensic Server.	Its reliance on proprietary cloud environments restricts its general applicability to public cloud deployments.

3.	Design and implementation of FROST: Digital forensic tools for the OpenStack cloud computing platform, by: Dykstra and Sherman	Visual Disks, API logs	Openstack cloud platform for cyber incident response.	Suggested a set of forensic tools for OpenStack cloud platform, namely: Virtual Disk Evidence Collector, Cloud API Activity Tracker, and Firewall Log Forensic Analyzer.	It had limited compatibility with OpenStack platforms presents challenges for broader applicability across heterogeneous cloud infrastructures.
4.	Multi-agent Systems for Dynamic Forensic Investigation, by: Philip et al.	DNS logs	Multi-agent system with decentralised Model	Developed a multi-agent forensic model where many agents work together to gather and analyse evidence from devices located in different places.	Decentralised agents increase system complexity, coordination overhead, communication delays, and risk of inconsistent or incomplete evidence collection across distributed devices.
5.	Artificial intelligence based digital forensics framework by, P.H. Rughani	Disk Images	Automate the Acquisition, Analysis and Presentation of Data for forensics	Framework to optimise speed and performance in investigating cyber crimes and minimising user interactions.	It remains unclear how the framework would address the handling of entirely new types of cybercrime that are not included in its training data.
6.	A novel malware detection system based on machine learning and binary visualization by, Baptista et al.	Binary files (Malicious and Benign files)	Malware detection based on binary visualisation using Neural Networks.	Describe a new approach to malware detection that combines machine learning with a creative method of visualising malware as images.	While their method reports promising accuracies for file types such as PDFs and DOCs, its generalizability to other malware formats and the computational cost of image processing for real-time

					applications remain uncertain.
7.	IoT security and the role of AI/ML to combat emerging cyber threats in cloud computing environment by, Temuchu et al.	Log files from CAIDA and Packt.	Data pre-processing, feature extraction using CNNs, and classification using SVMs.	Proposed a Hybrid Machine Learning (ML) approach for anomaly detection in IoT and cloud environments using Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs) to address security threats.	High computational cost and dependence on large, well-labelled datasets, making real-time deployment in resource-constrained IoT environments difficult.
8.	Reinforcement learning for an efficient and effective malware investigation during cyber incident response by, Dunsin et al.	Standard benchmark malware datasets	Explored the use of reinforcement learning (RL) to enhance malware investigation during cyber incident response.	Approach focuses on optimizing forensic decision-making through RL-based automation, improving efficiency and response times.	Challenges such as the explainability of RL models, computational costs, and scalability in real-world deployments remain underexplored.
9.	RCInvestigator: Towards better investigation of anomaly root causes in cloud computing systems by, S. Liu et al.	Cloud audit logs	Interactive framework that transforms cloud audit logs into a knowledge graph	Allowed experts to manually explore and annotate the root causes of anomalies through a multi-stage, human-machine collaborative process.	Depends heavily on expert involvement for its reasoning steps.
10.	**Our Project**	Network Traffic, HTTP Server Logs, .exe files	Real-Time feature Engineering for Classification with Docker containers and Kubernetes in cloud environments using deep learning model to reduce false positives.	System with multiple applications deployed to defend against cyber threats and respond to incidents. The system can interact with large amounts of data by scaling and predicting with higher accuracy.	Discussed in Gap Identification Section.

7. Gap Identification

i. One significant limitation is that while the system has been tested against a range of known attack types, it has yet to be rigorously evaluated under more comprehensive and diverse real-world threat scenarios. This includes handling the uncertainties associated with dynamic and evolving cyber threats, which may behave unpredictably compared to controlled environments.

ii. While dealing with anomaly detection in web applications, distinguishing between what constitutes normal behaviour and what qualifies as an anomaly becomes increasingly complex and ambiguous as the volume of data grows.

iii. The detection of outliers, especially when processing large datasets, remains a challenge.

iv. The occurrence of false positives is also possible, where the system incorrectly flags benign activities as malicious, which can lead to unnecessary alarms and impact the system's overall accuracy.

8. System Design and Development

The project consists of a three-tier architecture for efficient cyber threat detection and investigation. This architecture leverages containerisation technology to isolate and deploy various functionalities across three distinct environments: Production, Honeypot and DFIR. (Digital Forensics and Incident Response)

a) The Network Traffic Classifier

- Network traffic is a rich source of data that contains valuable information on the ongoing activity of the network.
- The logical design unfolds through three interconnected components, each contributing crucial functionalities to the overall system.
- Firstly, the packet capture engine serves as the foundational component, leveraging the Scapy library to capture network packets continuously. Operating within its designated container, this engine listens on specified interfaces, intercepts network traffic, and stores captured packets in PCAP files for subsequent analysis.
- Secondly, the packet analysis module, encapsulated within another container, reads the captured PCAP files, extracts connection-based statistics, and transforms them into structured datasets suitable for predictive modelling. Using the pre-trained machine learning model during the development phase, it predicts predefined attack labels.
- Finally, integrated into a separate container, the alerting mechanism monitors prediction outcomes and triggers alerts in real-time upon detecting anomalous network behaviour. These alerts serve as actionable insights for security analysts, enabling timely responses to potential threats. This containerised architecture ensures scalability, flexibility, and isolation, facilitating seamless deployment,

management, and scalability of the network traffic classifier application within diverse computing environments.

b) <u>Web Intrusion Detection System</u>

- The key novelty of this design lies in its real-time deployment and distributed data collection.
- Each web server can deploy a lightweight agent responsible for collecting and forwarding logs promptly to the Web Intrusion Detection System (WIDS).
- This agent-based approach ensures efficient log collection and minimises the impact on individual web servers.
- The WIDS utilises a shared volume accessible by the virtual machine (VM) and the deployed container. This shared volume facilitates efficient storage and access to the collected logs for real-time analysis.
- This application prioritises simplicity and efficiency by using a single container. The container performs the following tasks:
  - • Reads HTTP access logs from web servers.
  - • Pre-processes the log data by converting it to a pandas data frame for manipulation in Python.
  - • Generates features mentioned in Table 4 for anomaly detection from the log data.
  - • Sends the extracted features to the trained Isolation For est model for real-time anomaly detection.
- To reduce false positives, the application triggers alerts only when the number of detected anomalies exceeds a pre-defined threshold.
- This approach assumes that real-world attacks often involve rapid bursts of activity, leading to a surge in detected anomalies.

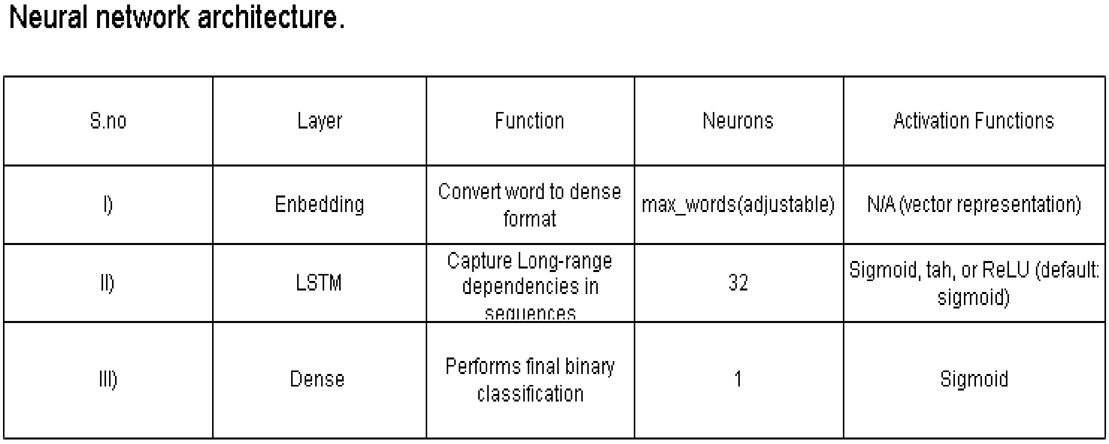
c) <u>Malware Analysis System</u>

- Malware analysis, which encompasses the investigation of malicious software's functionality, purpose, origin, and potential impact, traditionally requires extensive manual effort and expertise in software internals and reverse engineering. 3

<u>Deep Learning Model – Working Principle :</u>

- The deep learning model employs a Long Sh k, a type of Recurrent Neural Net work (RNN) w ata like text as explained below.

Keras model parameters

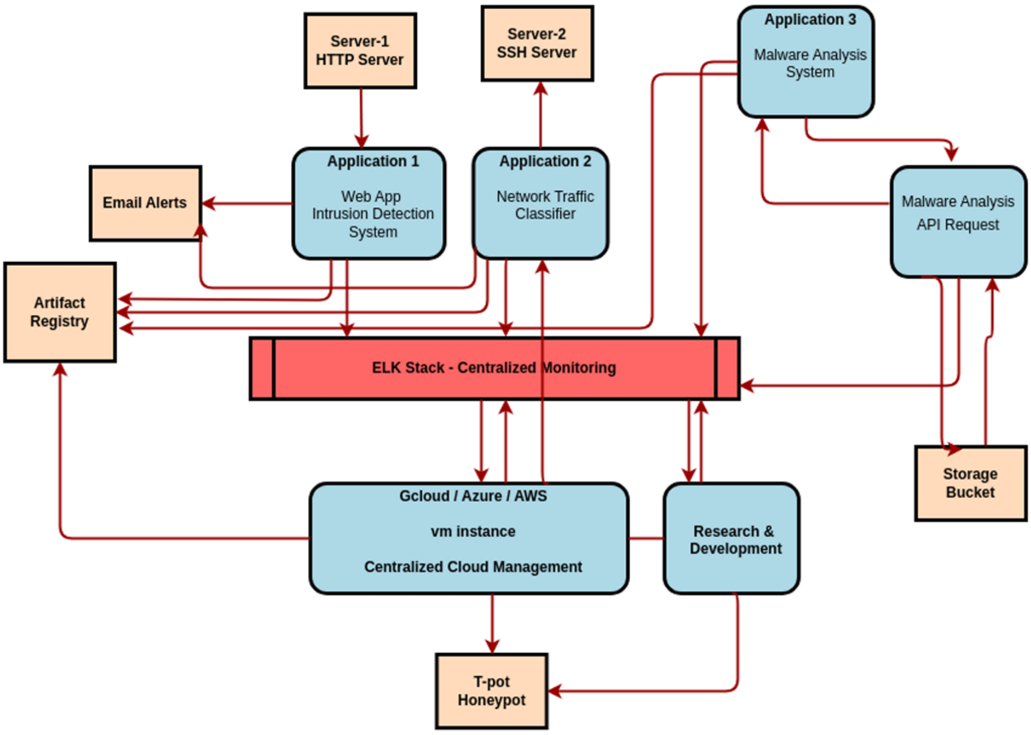


Neural network architecture.

S.no	Layer	Function	Neurons	Activation Functions
I)	Enbedding	Convert word to dense format	max_words(adjustable)	N/A (vector representation)
II)	LSTM	Capture Long-range dependencies in sequences	32	Sigmoid, tah, or ReLU (default: sigmoid)
III)	Dense	Performs final binary classification	1	Sigmoid

- The below table summarises the architecture of the neural network used in the malware analysis system, detailing each layer's function, the number of neurons, and the activation functions applied.
- The model begins with an embedding layer that converts discrete words into dense vector representations (32 dimensions in this case).
- An LSTM layer with 32 hidden units then processes these sequences. Unlike the embedding layer, each hidden unit within the LSTM can be considered a ''neuron'' that learns and extracts features from the data as summarised in the table above.
- Finally, a dense layer with a single output neuron and a sigmoid activation function performs binary classification. This single output neurone leverages the features learnt by the LSTM layer to predict the maliciousness of the analysed file

Proposed system data flow diagram



- The data flow diagram presented in the above diagram shows the connectivity and multiple pipelines implemented in the system.
- The system design allows for centralised management through the cloud infrastructure, enabling efficient monitoring and analysis.
- This architecture supports ongoing research and development, contributing to improved threat detection and response capabilities

9. Architecture Diagram , PTO.

- The system, as illustrated above, follows a logical flow that begins when the user uploads a file through the application's user interface.
- Upon submission, the uploaded file undergoes initial processing, where its features are extracted and prepared for classification.
- Using a pre-trained machine learning model, the system predicts the probability of the file being malicious.
- If the probability exceeds a threshold of 0.7, indicative of a high likelihood of malicious content, the system immediately classifies the file as ''Malicious'' and generates a detailed analysis report in PDF format. In contrast, if the probability falls between 0.5 and 0.7, the system invokes a Keras deep learning model for further binary classification.
- Upon completion of the classification, the system produces a comprehensive analysis report, facilitating informed decision making regarding the security implications of the file. For files deemed benign based on the classification results, the system provides a ''Benign''

10. Project Deliverables

i. Primary Deliverables:

a) An AI-Driven Cyber Incident Detection Framework, with a well-defined system architecture integrating network traffic classification, web intrusion detection, and malware analysis in cloud environments.

b) Trained and Evaluated ML/DL Models using benchmarked datasets, achieving high accuracy, precision, and recall across network, web, and malware threats.

c) A containerised security system deployed on cloud platform supporting scalable, real-time traffic, log and file analysis.

d) Malware classification system that generates probability-based predictions, and providing a detailed PDF reports or JSON files.

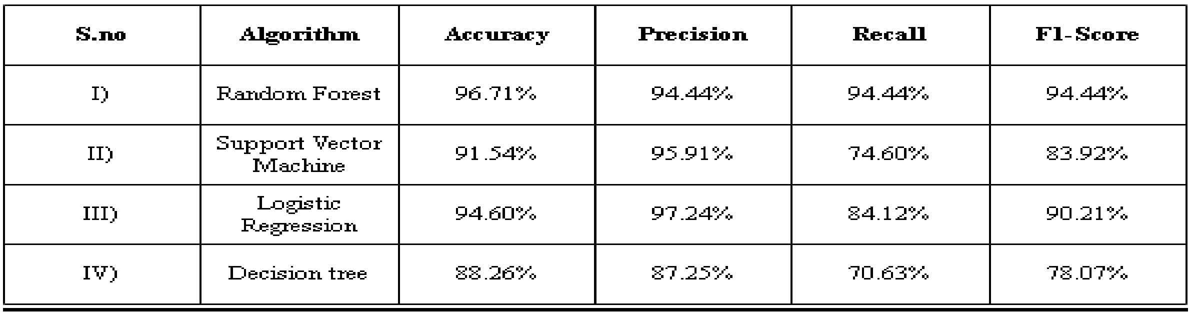
ii. Secondary Deliverables

a) A secure multi-VPC Cloud Architecture with isolated production, DFIR, and Honeypot VPCs with secure communication enabling safe forensic analysis.

b) An ELK Stack-based centralised logging, visualisation and alerting for incident response and monitoring.

c) Design of a Honeypot environment that captures real-world attack data to facilitate continuous learning and model improvement.

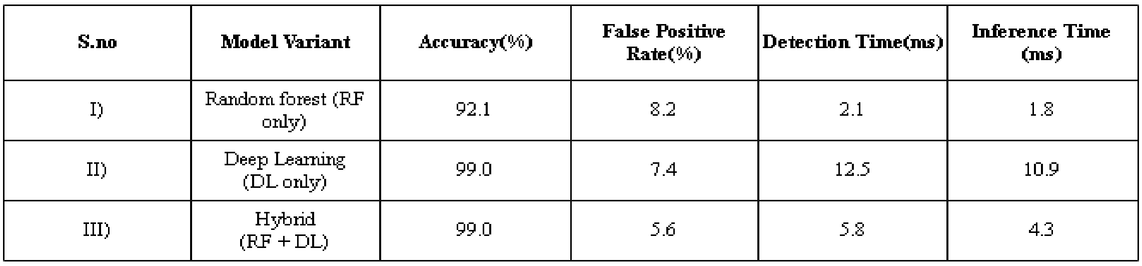
11.Performance Metrices



S.no	Algorithm	Accuracy	Precision	Recall	F1-Score
I)	Random Forest	96.71%	94.44%	94.44%	94.44%
II)	Support Vector Machine	91.54%	95.91%	74.60%	83.92%
III)	Logistic Regression	94.60%	97.24%	84.12%	90.21%
IV)	Decision tree	88.26%	87.25%	70.63%	78.07%

- The Random Forest algorithm exhibited an accuracy of 96%, while the Keras model achieved an even higher accuracy of 99%.
- Early testing revealed limitations in both models: Random Forest was computationally efficient but prone to false positives, whereas the Keras deep learning model demonstrated superior generalization but required significantly higher computational resources, leading to increased inference latency.
- To address these trade-offs, a hybrid approach was implemented, leveraging the strengths of both models.
- To validate its effectiveness, we conducted an ablation study comparing the contributions of Random Forest and Deep Learning individually, shown below.

Ablation study comparing individual and hybrid model performance.



S.no	Model Variant	Accuracy(%)	False Positive Rate(%)	Detection Time(ms)	Inference Time (ms)
I)	Random forest (RF only)	92.1	8.2	2.1	1.8
II)	Deep Learning (DL only)	99.0	7.4	12.5	10.9
III)	Hybrid (RF + DL)	99.0	5.6	5.8	4.3

- The hybrid model balances these strengths, reducing false positives while maintaining high accuracy and optimizing inference time.

12.Future Enhancements

- First, a more comprehensive set of attack scenarios needs to be incorporated into the testing set to enhance the overall system's robustness against real-world cyber threats.
- Continuous training of models using honeypot data will be crucial to keeping the system adaptable to new and emerging attack patterns. To support this, the exploration of TPU-powered VM instances will enable more efficient training of deep learning models within the DFIR environment, overcoming current computational constraints. Moreover, automating the deployment process will enable seamless system deployment across any cloud environment, whether AWS, Azure, or GCP, enhancing the system's flexibility and responsiveness.

- The use of this system design with multiple virtual private networks (VPNs) and Docker containers provides full control over the design while also supporting a multi-cloud approach

13.Conclusion

In this report, we have presented a systematic exploration and a feasible use of AI techniques under the umbrella of cyber security, emphasising on their seamless integration into incident response systems within cloud environments.

This has been achieved through the development and deployment of a cyber threat defence system, that includes a network traffic classifier application, malware analysis application, and web intrusion detection system.

By deploying the system on Google Cloud and Microsoft Azure, the scalability and versatility of AI-powered cybersecurity solutions were demonstrated. Testing on NSL-KDD, CIC-IDS-2017, UNSW-NB15, and VirusTotal datasets showed Random Forest achieving high accuracy, while deep learning improved precision. A modular, containerised architecture enabled real-time analysis, supported by T-Pot for continuous learning and the ELK Stack for log analysis and visualisation.

14.References

- [1] M. Ashfaaq M. Farzaan, M. Chahine Ghanem, A. El-Hajjar, and D. N. Ratnayake, ‘‘AI-enabled system for efficient and effective cyber incident detection and response in cloud environments,’’ 2024, arXiv:2404.05602.
- [2] D. Dunsin, M. C. Ghanem, K. Ouazzane, and V. Vassilev, ‘‘A com prehensive analysis of the role of artificial intelligence and machine learning in modern digital forensics and incident response,’’ Foren sic Sci. Int., Digit. Invest., v
- [3] J. Dykstra and A. T. Sherman, ‘‘Design and implementation of FROST: Digital forensic tools for the OpenStack cloud computing platform,’’ Digit. Inve
- [4] A. Zewdie and T. Girma, ‘‘IoT security and the role of AI/ML to combat emerging cyber threats in cloud computing environment,’’ Issues Inf. Syst., Issues I
- [5] C. Stelly and V. Roussev, ‘‘SCARF: A container-based approach to cloud scale digital forensic processing,’’ Digit. Invest., vol. 22, pp. S39–S47, Aug. 2017, doi: 10.1016/j.diin.2017.06.008.
- [6] N. Moustafa and J. Slay, ‘‘UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),’’ in Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS), Canberra, ACT, Australia, Nov. 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.

- [7] S.NandaandR.A.Hansen,“Forensicsasaservice:Three-tierarchitecture for cloud based forensic analysis,” in Proc. 15thInt.Symp.ParallelDistrib. Comput. (ISPDC), Jun. 2016, pp. 178–183, doi: 10.1109/ISPDC.2016.31.
- [8] P.Kendrick, A.Hussain, andN.Criado,“Multi-agent systems for dynamic forensic investigation,” in Intelligent Computing Theories and Appli cation, D. S. Huang, V. Bevilacqua, and P. Premaratne, Eds., Cham, Switzerland: Springer, Jan. 2016, pp. 796–807, doi: 10.1007/978-3-319 42291-6_79.
- [9] I. Baptista, S. Shiaeles, and N. Kolokotronis, “A novel malware detection systembasedonmachinelearningandbinaryvisualization,”inProc.IEEE Int. Conf. Commun. Workshops (ICC Workshops), May 2019, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8757060
- [10] P. H. Rughani, “Artificial intelligence based digital forensics framework,” Int. J. Adv. Res. Comput. Sci., vol. 8, no. 8, pp. 10–14, Oct. 2017, doi: 10.26483/ijarcs.v8i8.4571.
- [11] (1999). NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB. [Online]. Available: https://www.unb.ca/cic/ datasets/nsl.html
- [12] A. Boukhamla and J. C. Gaviro, “CICIDS2017 dataset: Performance improvements and validation as a robust intrusion detection system testbed,” Int. J. Inf. Comput. Secur., vol. 16, no. 1, p. 20, 2021, doi: 10.1504/ijics.2021.117392
- [13] DeutscheTelekomSecurityGmbH(2024).T-Pot24.04.0(Version24.04.0) [Computer Software]. [Online]. Available: https://github.com/telekom security/tpotce
- [14] A. Bhardwaj and K. Kaushik, “Predictive analytics-based cybersecurity framework for cloud infrastructure,” Int. J. Cloud Appl. Comput., vol. 12, no. 1, pp. 1–20, 2022, doi: 10.4018/IJCAC.297106.
- [15] S.Liuetal.,“RCInvestigator: Towards better investigation of anomaly root causes in cloud computing systems,” 2024, arXiv:2405.15571.
- IEEE Journal: AI-Powered System for an Efficient and Effective Cyber Incidents Detection and Response in Cloud Environments, by Mohammed Ashfaaq M. Farzaan, Mohamed Chahine Ghanem, Ayman El-Hajjar and Deepthi N. Ratnayake