AI Predictions In Cyber Security Analysis and Early Detection Of Cyberattacks

Soulyma Al-Mahaeri

Abstract

The cyber-attack landscape has evolved as a result of the increased prevalence of digitalization and interconnected systems. This necessitates the development of innovative methods to understand and counter these threats effectively. This research explores the connection between cybersecurity and artificial intelligence, emphasizing how AI can improve the detection of cyber-attacks through evaluation, prediction, and other approaches. Utilizing machine learning algorithms, data analytics predictive models driven by AI have emerged as a way to address the constantly evolving challenges presented by cyber threats.

The primary objective of this study is to assess the performance of AI-driven prediction in the field of cyber security. The focus is on examining the effectiveness of these AI-based systems in comparison to traditional cyber security approaches, with an emphasis on their ability to proactively identify and address cyber threats in order to minimize their impact. Furthermore, we also explore the constraints and ethical considerations related to AI-based solutions for cyber security. Additionally, it involves utilizing AI algorithms for analyzing and early detection of Cyber Attacks through machine learning algorithms developed using Python programming language.

The findings of the study hold significant implications for the field of cyber security, offering insight into the future trends in threat mitigation and incident response. This research contributes to the advancement of cutting-edge cyber security solutions by addressing the constantly changing landscape of cyber threats. Additionally, it fosters comprehension of ethical and regulatory aspects associated with incorporating AI in combating cyber-attacks. The convergence of cyber security and artificial intelligence constitutes a vital area of inquiry in this era of digital transformation, essential for preserving the integrity and security of our interconnected global society.

KEYWORDS: Cyber security, Cyber Attacks, Artificial Intelligence, Data Analysis, Machine Learning, Analysis, Early Detection, Predictive Models

Table of Contents

Table of Tables

Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
APT	Advanced Persistent Threat
DoS	Denial of Service
NSA	National Security Agency
AIS	Artificial Immune System
R2L	Remote-to-Local
U2R	User-to-Root
CPS	Cyber-Physical Systems
SDN	Software-Defined Networking
IoT	Internet of Things
AIoT	Artificial Internet of Things
SIEM	Security Information and Event Management
KNN	K-Nearest Neighbor
SVM	Support Vector Machine
API	Application Programming Interface
DNS	Domain Name System
IP	Internet Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
CVE	Common Vulnerabilities and Exposures
OSINT	Open-Source Intelligence
IOC	Indicators of Compromise
SOC	Security Operations Center

C2C Command and Control

ROI Return on Investment

IoC Indicators of Compromise

MITM Man-in-the-Middle

Introduction

In the contemporary era of digital advancement, the unprecedented global connectivity facilitated by technological progress has ushered in a plethora of opportunities and conveniences. However, this interconnectedness has also exposed societies and systems to the heightened risk of cyber-attacks, which continuously evolve to exploit vulnerabilities within digital infrastructure. These attacks extend beyond the realms of data and systems, posing tangible threats to individuals, organizations, and nations. Consequently, the protection of digital assets through robust cybersecurity measures has become an indispensable defense mechanism in this dynamic digital landscape

The evolving landscape of cyber threats necessitates a continuous evolution of cybersecurity strategies to effectively counter emerging attack vectors and sophisticated adversaries. The rapid pace at which threats mutate challenges conventional security paradigms, highlighting the need for adaptive and innovative security solutions.

The contemporary era is characterized by a proliferation of cyber dangers, marked by the emergence of state-sponsored cyber operations, the global reach of cyberwarfare, and the increasing complexity of attacks. This study aims to underscore the imperative of early threat detection and preemptive measures in safeguarding cyber infrastructure and data integrity. Through the analysis of real-world cases, this research illuminates the efficacy of artificial intelligence (AI) in enhancing cybersecurity efficacy, particularly in scenarios where early detection and predictive models have thwarted significant security breaches.

However, the integration of AI in cybersecurity also raises profound concerns regarding privacy, ethical considerations, and biases inherent in AI systems. These apprehensions underscore the necessity of ensuring that the benefits of AI-driven cybersecurity initiatives are accompanied by a commitment to ethical standards and accountability.

The realm of artificial intelligence stands at the vanguard of cybersecurity evolution, offering scalable and adaptive solutions that leverage self-learning capabilities to counter new and evolving threats effectively. The proactive nature of AI enables the development of agile security strategies tailored to contemporary cybersecurity challenges. This study explores the synergies between Artificial Intelligence Forecasting in Cybersecurity and the Early Detection of Cyber Attacks, seeking to elucidate the potential of AI in predicting, detecting, and mitigating cyber threats.

By harnessing predictive analytics, automating threat detection, and adapting swiftly to emerging attack methodologies, AI has the capacity to revolutionize cybersecurity practices by augmenting traditional defense mechanisms with advanced predictive capabilities. This study aims to assess the efficacy of AI-driven predictive models in enhancing cybersecurity frameworks, benchmarking their performance against conventional approaches, and addressing potential limitations or concerns associated with their deployment.

Chapter 1: Project Description

1.1 Problem Statement

In the contemporary digital landscape, characterized by widespread technology adoption, the escalating threat of cyber-attacks poses a significant challenge for individuals, organizations, and nations. Developing innovative and effective cybersecurity solutions is imperative due to the intricate nature and increasing sophistication of these attacks. Artificial intelligence (AI) has emerged as a transformative technology in the realm of cybersecurity, offering promising capabilities in predicting, analyzing, and enhancing early detection of cyber threats. However, leveraging AI to its full potential in bolstering cyber defenses remains a critical endeavor.

This research endeavors to address the urgent demand for bolstering cybersecurity defenses in the face of escalating and unpredictable cyber-attacks. Conventional cybersecurity methodologies, albeit essential, often adopt reactive approaches focused on damage control post-attack, leading to significant delays in intrusion detection. This delay provides malicious actors with ample opportunity to exploit vulnerabilities and inflict substantial harm. By embracing proactive and predictive defense mechanisms through AI integration, the cybersecurity landscape stands to benefit from enhanced resilience against such threats.

AI harnesses advanced machine learning algorithms, neural networks, and data analysis tools to identify patterns, anomalies, and emerging risks proactively. This proactive approach, underpinned by predictive modeling, has the potential to revolutionize data asset protection and fortify digital infrastructure.

This study aims to delve into the role of artificial intelligence in cybersecurity and its efficacy in early cyber threat identification by exploring the following key questions:

- How can AI-driven predictive models enhance early detection and analysis of cyber-attacks?
- What impact does AI have on cybersecurity effectiveness compared to traditional methodologies?
- What are the challenges and implications associated with utilizing AI for cybersecurity predictions?

By unraveling valuable insights throughout this exploration, we seek to illuminate the potential benefits of AI in cybersecurity and foster advancements in this critical domain to bolster the security of our digital environment.

1.2 Project Objectives

The primary aim of this research is to meticulously investigate the integration of artificial intelligence (AI) in cybersecurity, focusing specifically on AI's capabilities in predicting, analyzing, and aiding in the early detection of cyber-attacks. The project objectives derived from this overarching goal can be outlined as follows:

1. **Assessing the Efficacy of AI-Powered Predictive Models:** This research aims to thoroughly investigate the efficacy of predictive models based on artificial intelligence in enhancing early detection of cyber-attacks. The study will assess how well AI systems can identify, react to, and alleviate developing threat patterns, taking into account measures like precision, effectiveness, and swiftness.
2. **Comparison with Traditional Methods:** A key objective of this research is to compare the performance of AI-driven cybersecurity solutions with traditional methodologies. By examining the outcomes and capabilities of AI systems against conventional practices, the aim is to determine the extent to which AI enhances the early detection and analysis of cyber threats.
3. **Showcasing Real-Life Applications:** This study aims to provide valuable insights by examining real-world case studies that highlight scenarios where AI-driven predictive models have played a critical role in preventing significant security breaches. By demonstrating successful implementations, the goal is to underscore the tangible benefits of AI in strengthening cybersecurity.
4. **Identifying Future Trajectories and Research Areas:** The objective of this research is to identify key areas for future exploration and advancement. By examining the dynamic landscape of cyber threats, the study aims to propose strategic avenues for enhancing AI-driven cybersecurity, addressing emerging challenges, and leveraging interdisciplinary approaches to bolster defenses.

In essence, the research's objectives align with the overarching goal of deepening our understanding of integrating artificial intelligence into cybersecurity practices. Through a thorough pursuit of these objectives, the research aims to contribute to the development of cutting-edge cybersecurity solutions and promote the ethical and responsible deployment of AI in the ever-evolving cyber threat landscape.

1.3 Project Features

As global entities, businesses, and individuals strive to combat the escalating threat of cyberattacks, the integration of artificial intelligence (AI) into cybersecurity presents a multitude of advantages that have the potential to redefine the cybersecurity landscape. These features span various facets and have the capacity to transform cybersecurity practices:

1. **Enhanced Early Detection and Prevention:** A pivotal contribution of AI to cybersecurity lies in its ability to predict, analyze, and facilitate early detection of cyber threats. AI-driven predictive models excel at identifying patterns and anomalies, enabling swift detection and mitigation of threats before they can inflict significant damage. This capability is crucial for safeguarding data integrity and digital assets.
2. **Proactive Threat Response:** AI equips cybersecurity systems with proactive threat response capabilities. By continuously learning from and adapting to emerging threats, AI can autonomously respond to novel and unforeseen attack vectors. This proactive stance reduces the susceptibility to new forms of cyberattacks.
3. **Reduction in False Positives:** Leveraging cutting-edge algorithms, AI-powered predictive models significantly reduce the occurrence of false positives in threat identification. This reduction enhances the operational efficiency of cybersecurity strategies by directing human attention towards real threats instead of inundating analysts with false alerts.
4. **Promptly Addressing Incidents:** In the dynamic realm of cyber threats, the ability of AI to expedite incident response times is critical. Automated response systems enabled by AI facilitate real-time thwarting of attacks, leading to quicker incident resolution and mitigating potential impacts.
5. **Adaptability to Changing Threats:** Artificial Intelligence stands out for its adaptability in combating the constantly evolving cyber threat landscape. Its agility and versatility enable swift adaptation to new attack methodologies and vulnerabilities, ensuring the enduring efficacy of cybersecurity measures.
6. **Enhanced Data Privacy and Protection:** AI plays a pivotal role in enhancing data privacy by promptly identifying and addressing data breaches. It can aid organizations in adhering to data security regulations, pinpointing vulnerabilities, and enhancing data protection protocols.
7. **Resource Efficiency and Cost Savings:** By reducing the reliance on extensive human monitoring and intervention, AI-driven cybersecurity models offer cost savings. These systems can process large volumes of data swiftly, enabling more efficient resource allocation for organizations.

Chapter 2: Literature Review

2.1 Related works

2.1.1 A blockchain-based cyber-attack detection scheme for decentralized Internet of Things using software-defined network [34]

This study presented a decentralized framework created to identify and address various security breaches within an Internet of Things environment, with the goal of enhancing security measures. This approach goes beyond previous models in the field and introduces three key innovations: first, it utilizes Software-Defined Networking for ongoing monitoring and analysis of data traffic in the IoT context, enabling early detection and mitigation of attacks; secondly, it employs blockchain technology to establish a decentralized architecture that overcomes the single point failure limitation of earlier centralized and distributed architectures in this area [34].

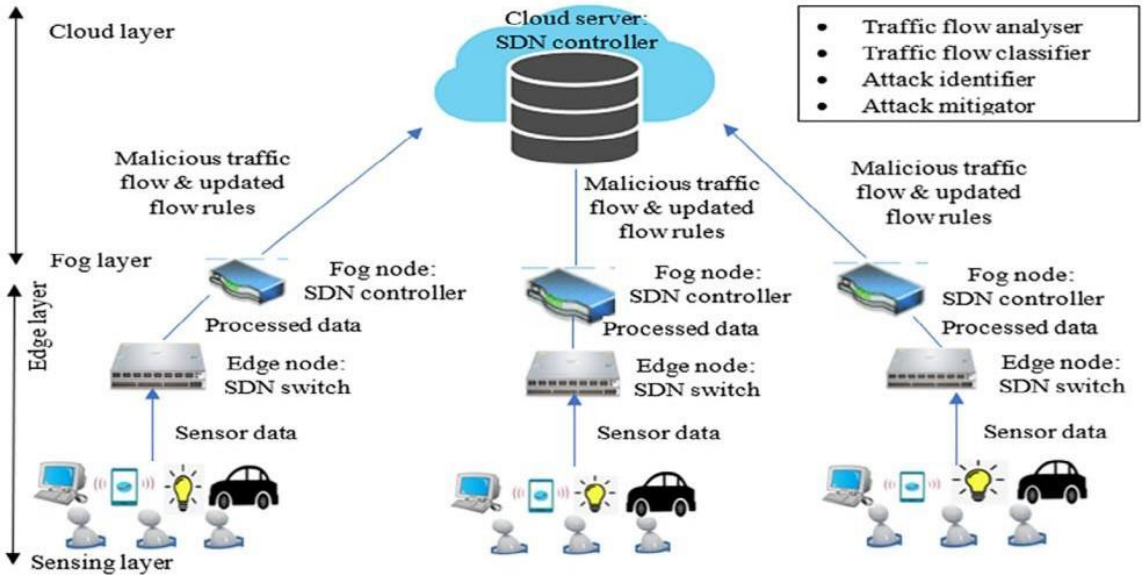
The structure consists of three layers. The fog nodes play a key role in detecting attacks and communicating protocol changes to the edge nodes, which leads to shorter incident response and recovery times. Research results demonstrate that this method is highly effective, outperforming current approaches in the field and exhibiting potential as an essential security component for upcoming Internet of Things networks.

The following figure demonstrates the identification of harmful actions within the system. The cooperative and decentralized attack detection system integrates the Cloud, Fog, Edge, and Sensing layers, providing an outline of the approach. Each fog node has full authority over the attack detection mechanism.

Detection devices produce significant data that is transferred to the edge layer, the following level, for surveillance and identification of events within their specific surroundings. Each edge layer consists of advanced SDN-enabled switches that support multiple sensors below them, tasked with managing traffic and analyzing it at the connected sensory nodes.

The fog layer receives analyzed information from the edge layer. Clusters of SDN switches form the fog nodes, which identify abnormal traffic patterns within their connected nodes. By adjusting traffic regulations when unusual activities are detected, SDNs teach their switches to swiftly detect and respond to attacks. Data from all SDN controllers is then sent to the cloud for additional examination of network activity, security threats, and unusual behaviors.

Sensory nodes detect and monitor traffic, while edge nodes analyze and manage data. SDNs at the fog nodes learn traffic patterns to identify malicious actions within the system. Different attributes are employed to examine historical traffic patterns, allowing for detection of known attacks and identification of anomalous behavior in nodes. The SDN controller creates rules for managing traffic and distributes them dynamically to the applicable switches.



Detection of malicious behaviors of the system

Upon completion of data collection by sensory nodes and traffic monitoring, the gathered data is then analyzed and managed by edge nodes. Additionally, SDNs at fog nodes learn about traffic patterns to detect potential malicious activities. Various characteristics are utilized to analyze historical traffic patterns in order to identify known attacks and determine whether a node is exhibiting normal or anomalous behavior. The SDN controller creates traffic rules and communicates them dynamically to relevant switches. These rules, based on the evaluation by the SDN controller, may block specific traffic and restrict flow in cases of abnormal behavior. Consequently, the SDN takes on the responsibility for monitoring all network activities and making decisions for interconnected nodes. In order to create a distinct attack detection model for each fog node, key components of the SDN controller including the traffic analyzer, traffic classifier, attack identifier, and attack mitigator are assigned with identifying malicious activity within network traffic. As predefined time intervals are adjusted over time,the blockchain technique is eventually employed as a means to mitigate attacks targeting edge nodes.

2.1.2 Secure and resilient artificial intelligence of things [35]

The researchers of this study introduce a HoneyNet approach that integrates threat detection and situational awareness to enhance the security and resilience of AIoT. They initiate the process by establishing a HoneyNet utilizing Docker technology to collect data for threat identification and monitor attack methodologies. Subsequently, the gathered data is converted into images and employed as training data for a deep learning model.

Ultimately, AIoT leverages the trained model to provide threat identification and situational awareness services. To validate this approach, the researchers employ the SiteWhere AIoT platform to deploy HoneyNets, train models, and create a threat simulation environment for both detection and situational awareness on this platform. The test results demonstrate the effectiveness and viability of this solution.

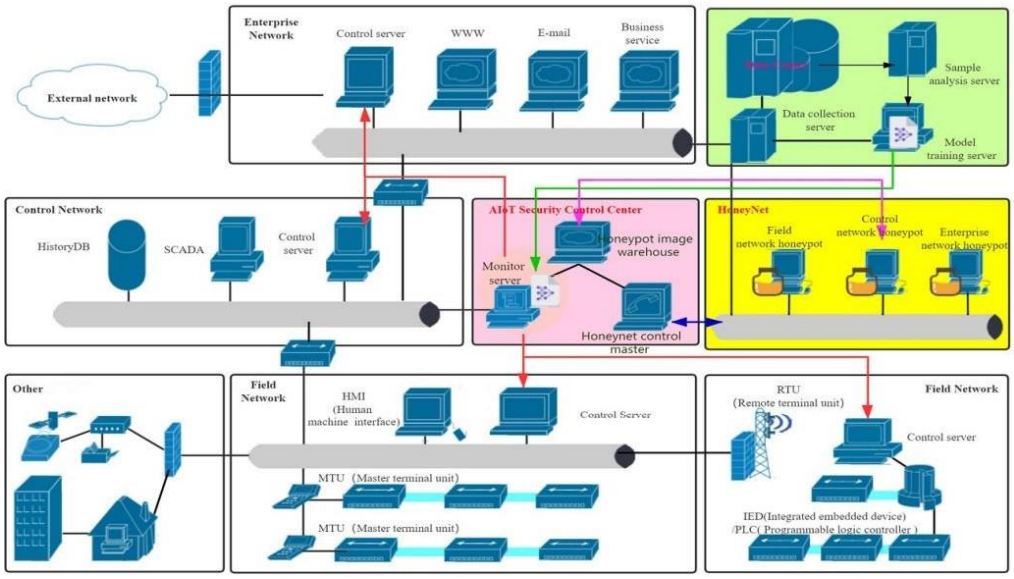
The following Figure shows illustrates the network architecture of the detection model, enhancing the Industrial Internet of Things (IIoT) standard architecture with three new components. These components are detailed as follows:

1. ASCC (AIoT Security Control Center): This serves as the core element. The ASCC's primary functions encompass situational awareness, security threat monitoring, and HoneyNet management. The ASCC comprises the following three nodes:
   - Honeynet Control Master (HCM): Responsible for communication and supervision of HoneyNet nodes, ensuring the coherent operation of the honeypot system.
   - Honeypot Image Warehouse (HIW): Organizes and stores honeypot images.
   - Monitor Server (MS): Interfaces with control servers across enterprise and field networks of AIoT to detect threats promptly and maintain situational awareness using a deep learning model.

2. HoneyNet: Dedicated to detecting harmful applications and malicious activities within AIoT, the HoneyNet includes three types of honeypots:
   - Enterprise Network Honeypot (ENH): Detects and mitigates malicious activities within the enterprise network.
   - Control Network Honeypot (CNH): Identifies and addresses malicious activities within the control network.
   - Field Network Honeypot (FNH): Targets malicious behavior within the field network.

3. Data Center (DC): Responsible for data collection and preprocessing, the DC feeds information such as malicious behavior and HoneyNet data into a deep learning CNN model. The DC comprises two key nodes:
   - Data Collection Servers (DCS): Gather information and backup source logs in case of honeypot data loss.
   - Sample Analysis Server (SAS): Compiles and integrates logs from various honeypots, storing the data using a Hadoop Distributed File System.

Key data features extracted for analysis include ports, numbers, attacker IP addresses, attack pathways, requested content, vulnerability fingerprints, and malicious sample server IP addresses. Additionally, the Model Training Server (MTS) focuses on training modules utilizing CNN for deep learning purposes.

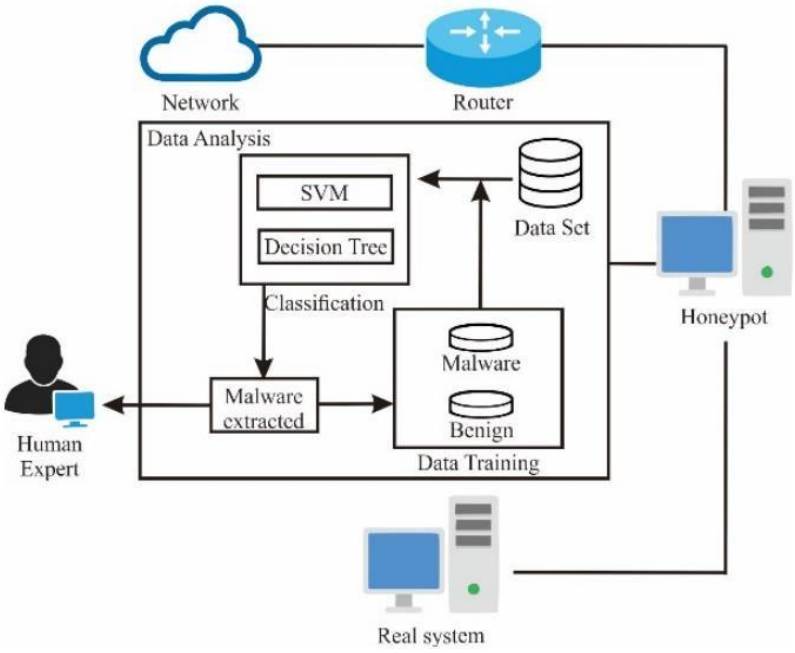


The network architecture for the detection model

The previous figure outlines the communication pathways connecting the MS with the field, control, and enterprise networks. The red line denotes this connection. The MS is responsible for distributing monitoring directives to all levels of control servers, while receiving real-time statuses from the field, control, and business networks. This information exchange is represented by green lines between MTS and MS. Specifically, the MS retrieves trained deep learning model parameters from MTS using data exchange indicated by a green line. Additionally, there's an information flow marked by a purple line that transfers HIW's mirror image to Honeypot signifying their interaction. Furthermore, there's an exchange of control information denoted by a blue line between honeypot and HCM where status updates are received regarding sample collection at HCM while honeypot receives control directives specifically for deep learning purposes

2.1.3 Malware detection using honeypot and machine learning [36]

The study proposed a framework for machine learning and honeypot-based malware detection, detailing the experimental protocols to be adhered to. Support Vector Machines (SVM) and decision algorithms were employed in the process. Findings indicated that decision and SVM algorithms exhibited superior performance in accuracy, detection rates, false alarms, and the precision of distinguishing four distinct types of attacks. The architectural design enabled the identification of malware based on behavioral patterns, with provisions for self-learning based on detection outcomes to uncover novel malware techniques.



The architecture of the used method to detect malware

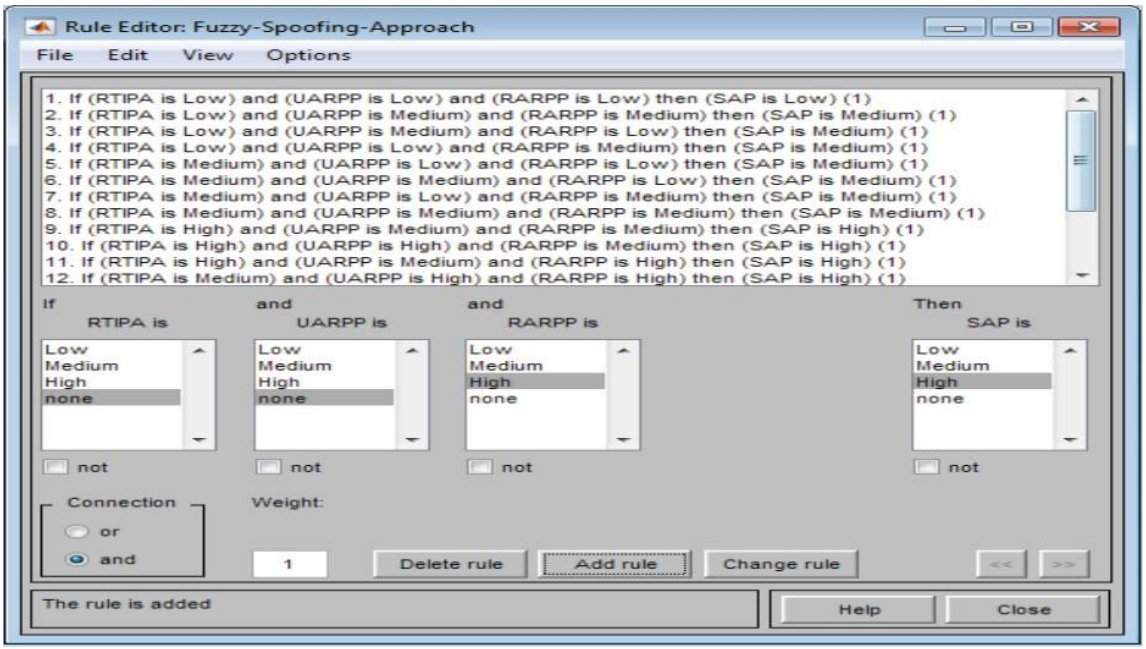
The figure depicts the framework employed for malware detection, showcasing a comprehensive architectural layout encompassing elements such as honeypots, machine learning modules, routers, data analysis components, and actual system entities. Incoming traffic from external networks traverses the router to reach the internal network, where packets are further directed to the honeypot. The honeypot captures and stores traffic packets entering the internal network on a simulated server, facilitating subsequent analysis. Within the data analysis module, a malware detection system leverages datasets for training and testing purposes. The subsequent section will delve into the dataset utilized for the study.

The classification process of the dataset employs the Decision Tree algorithm and Support Vector Machine (SVM) algorithm. The classification methodology is geared towards distinguishing between malicious and benign data classes. Upon identifying malware through the classification algorithm, the system extracts and reports the findings to human experts. Additionally, retraining occurs to enhance detection accuracy.

The algorithms have demonstrated optimal efficiency and accuracy in their performance. To maximize accuracy, a split-testing approach with a 90:10 ratio is adopted for training and testing datasets. Ten experimental runs are conducted to validate the testing process.

2.1.4 A fuzzy approach for detecting and defending against spoofing attacks on low interaction honeypots [37]

The investigation recommends an enhanced strategy using fuzzy logic to identify and stop spoofing attacks on low-interaction honeypots. The main objective is to create a technique for identifying spoofing attacks by examining empirical data from the honeypot and its internal network. Moreover, the research suggests utilizing fuzzy logic to anticipate and alert users about potential spoofing attacks on low-interaction honeypots, effectively minimizing such risks. Experimental modeling validates that employing the proposed fuzzy method allows any low-interaction honeypot to detect and prevent spoofing attempts.



The fuzzy rules used to predict the spoofing attacks

The literature review commences with an experimental assessment of the KFSensor low-interaction honeypot spoofing attack, followed by the proposition of a spoofing attack detection strategy founded on an analysis of test data obtained from the honeypot and its intrinsic network.

Subsequently, a fuzzy approach is recommended for anticipating and promptly alerting about a honeypot spoofing incursion to thwart it effectively. Ultimately, the deployment of the fuzzy method is showcased to illustrate how an experimental simulation can transform any low-interaction honeypot into an "attack-aware" system against spoofing attacks.

Given that low-interaction honeypots are engineered with limited resources to mimic targeted services, they exhibit vulnerability to certain prevalent attacks. Due to the susceptibility of low-interaction honeypots to being spoofed, a sophisticated system that optimizes resources is essential for identifying spoofing attempts and predicting the probability of a spoofing attack on the honeypot.

In response to this necessity, the fuzzy technique furnishes an intelligent and resource-efficient resolution grounded in the detection process. Fuzzy rules are formulated based on the three designated fuzzy input variables, with an illustration of a fuzzy rule presented in previous figure.

2.2 comparison between four case studies of early detection in cyber security system

All of the following studies contribute valuable insights and represent significant advancements in the field. However, it is essential to acknowledge their limitations, which is a key focus of our proposed strategy. Building upon these limitations, this research introduces a new and comprehensive security approach designed to address the shortcomings identified in prior studies while maximizing their advantages. Our proposed scheme aims to enhance protection against the evolving landscape of cyber threats by implementing a robust security framework.

The envisioned security framework leverages Artificial Intelligence (AI) and machine learning algorithms for the analysis and detection of cyber-attacks. This approach builds upon the methodologies employed in previous research, which demonstrated superior efficacy compared to conventional methods in detecting and analyzing various types of cyber threats. Our proposed model is designed to mitigate the shortcomings observed in previous studies while aiming to establish a comprehensive framework that encapsulates numerous benefits.

Reference Paper	Year	Objectives	Techniques used	Achievements	Limitations
A blockchain-based cyber-attack detection scheme for decentralized Internet of Things using software-defined network	2021	Developing a robust approach for detecting attacks, and the approach should address the obstacles encountered by current systems.	Blockchain technology.	A decentralized structure that can identify and reduce the likelihood of different security attacks in IoT environments.	Using blockchain technology in the field of cybersecurity is resource-intensive.
Secure and resilient artificial intelligence of things	2021	To address the issue that traditional approaches to network security may not be fully suitable for the Internet of Things.	Artificial Intelligence and Honeypots.	An approach using HoneyNet to enhance the security and resilience of the Internet of Things by integrating threat detection with situational awareness.	Using such a kind of AI approaches needs experts on the field.
Malware detection using honeypot and machine learning	2019	Utilizing artificial intelligence and decoy systems in a structural plan to detect malicious software.	High interaction Honeypots, machine learning techniques and algorithms.	A machine learning algorithm was utilized to achieve exceptionally high accuracy, yielding results with optimal efficiency.	The deployment of high-interaction honeypots results in significant time consumption as it requires actual physical equipment.
A fuzzy approach for detecting and defending against spoofing attacks on low interaction honeypots	2018	a method for detecting spoofing attacks by analyzing experimental data obtained from a honeypot and its internal network.	Fuzzy Logic	A fuzzy logic approach to optimize resources for detecting and preventing spoofing attempts on honeypots with minimal interaction	Depending on fuzzy logic rules in the domain of security and cybersecurity does not yield optimal outcomes due to their lack of precision.
Proposed work	2024	Machine learning has been applied in the field of cybersecurity to provide significant support and advantages.	AI and Machine learning	Machine learning techniques for analyzing and identifying cyber threats	It only suffers from minor limitations, which are that it needs to implement broader types of attacks and security threats

Table 1 Comparison between four case studies of early detection in cyber security system and the proposed work

Chapter 3: Research Methodology

3.1 Data Collection

The raw network packets of the UNSW-NB 15 dataset were created by the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviors.

Tcpdump tool is utilized to capture 100 GB of the raw traffic. This dataset has nine types of attacks, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. The Argus, Bro-IDS tools are used and twelve algorithms are developed to generate totally 49 features with the class label.

For further information check the link Dataset

3.1.1 Dataset features

1. script: Source IP address
2. sport: Source port IP
3. dstip: Destination IP address
4. dsport: Destination port number
5. proto: Transaction protocol
6. state: Indicates the state and its dependent protocol, e.g., ACC, CLO, CON, ECO, ECR, FIN, INT,MAS, PAR,...
7. dur: Record total duration
8. sbytes: Source to destination transaction bytes
9. dbytes: Destination to source transaction bytes
10. sttl: Source to destination time to live value
11. dttl: Destination to source time to live value
12. sloss: Source packets retransmitted or dropped
13. dloss: Destination packets retransmitted or dropped
14. service: http, ftp, smtp, ssh, dns, ftp-data, irc and (-) if not much used service
15. Sload: Source bits per second
16. Dload: Destination bits per second
17. Spkts: Source to destination packet count
18. Dpkts: Destination to source packet count
19. swin: Source TCP window advertisement value
20. dwin: Destination TCP window advertisement value
21. stcpb: Source TCP base sequence number
22. dtcpb: Destination TCP base sequence number
23. smeansz: Mean of the flow packet size transmitted by the source
24. dmeansz: Mean of the flow packet size transmitted by the destination
25. trans_depth: Represents the pipelined depth into the connection of http request/response transaction

26. res_bdy_len: Actual uncompressed content size of the data transferred from the server's http service
27. Sjit: Source jitter (mSec)
28. Djit: Destination jitter (mSec)
29. Smite: Record start time
30. Ltim: Record last time
31. Sintpkt: Source interpacket arrival time (mSec)
32. Dintpkt: Destination interpacket arrival time (mSec)
33. tcprtt: TCP connection setup round-trip time, the sum of 'synack' and 'ackdat'
34. synack: TCP connection setup time, the time between the SYN and the SYN_ACK packets
35. ackdat: TCP connection setup time, the time between the SYN_ACK and the ACK packets
36. is_sm_ips_ports: If source (1) and destination (3) IP addresses equal and port numbers (2)(4) equal then, this variable...
37. ct_state_ttl: No. for each state (6) according to specific range of values for source/destination time to live (10...
38. ct_flw_http_mthd: No. of flows that has methods such as Get and Post in http service
39. is_ftp_login: If the FTP session is accessed by user and password then 1 else 0
40. ct_ftp_cmd: No of flows that has a command in FTP session
41. ct_srv_src: No. of connections that contain the same service (14) and source address (1) in 100 connections...
42. ct_srv_dst: No. of connections that contain the same service (14) and destination address (3) in 100 connections...
43. ct_dst_Item: No. of connections of the same destination address (3) in 100 connections according to the last time...
44. ct_src_Itm: No. of connections of the same source address (1) in 100 connections according to the last time (26)...
45. ct_dtc_dport_Itm: No of connections of the same source address (1) and the destination port (4) in 100 connections acc...
46. ct_dst_sport_Itm: No of connections of the same destination address (3) and the source port (2) in 100 connections acc...
47. ct_dst_src_Itm: No of connections of the same source (1) and the destination (3) address in 100 connections according...
48. attack_cat: The name of each attack category. In this data set, nine categories e.g. Fuzzers, Analysis, Backdoo...
49. Label: 0 for normal and 1 for attack records

3.1.2 Data Analysis

For data analysis the following libraries were used:

1. **Pandas**: Pandas is a powerful data manipulation library in Python that provides data structures like DataFrames and Series. It offers a wide range of functions for data cleaning, transformation, and analysis.

2. **NumPy**: NumPy is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently.

3. **Seaborn**: Seaborn is a data visualization library based on Matplotlib. It offers a high-level interface for creating attractive and informative statistical graphics. Seaborn simplifies the process of creating complex visualizations by providing easy-to-use functions., the main following functions from seaborn were used:
   - Boxplot: This function in Seaborn is used to create boxplots, which display the distribution of a dataset along with statistical summaries. Boxplots are useful for identifying outliers, comparing distributions, and visualizing the spread of data.

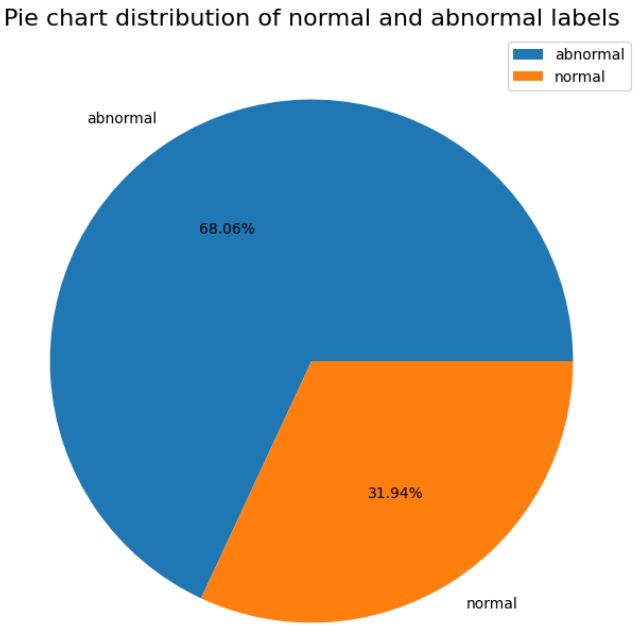
   - Heatmap: This function in Seaborn is used to create heatmaps, which visualize data in a tabular format where colors represent the values. Heatmaps are particularly useful for displaying correlation matrices, as they provide a quick and intuitive way to identify patterns and relationships between variables.

4. **Matplotlib**: Matplotlib is a comprehensive plotting library in Python that produces publication-quality figures. It enables the creation of a wide range of plots, including line plots, scatter plots, bar plots, histograms, and more.

3.2 Pre-Processing

3.2.1 Binary Classification

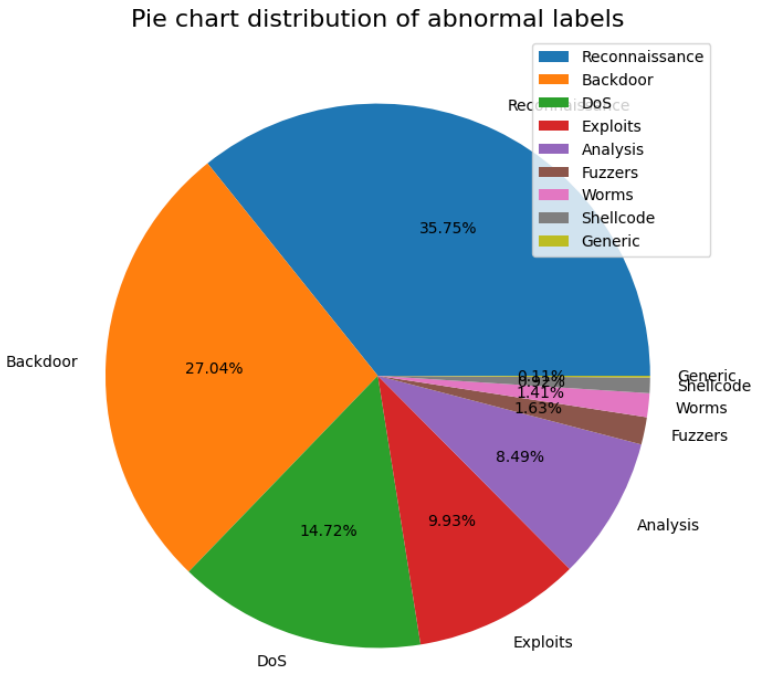
To develop a binary classification model aimed at identifying whether data entries represent normal or abnormal activities, the specific types of attacks are irrelevant. Therefore, the preprocessing step involved solely removing the categorical "cat-attack" column from the dataset.



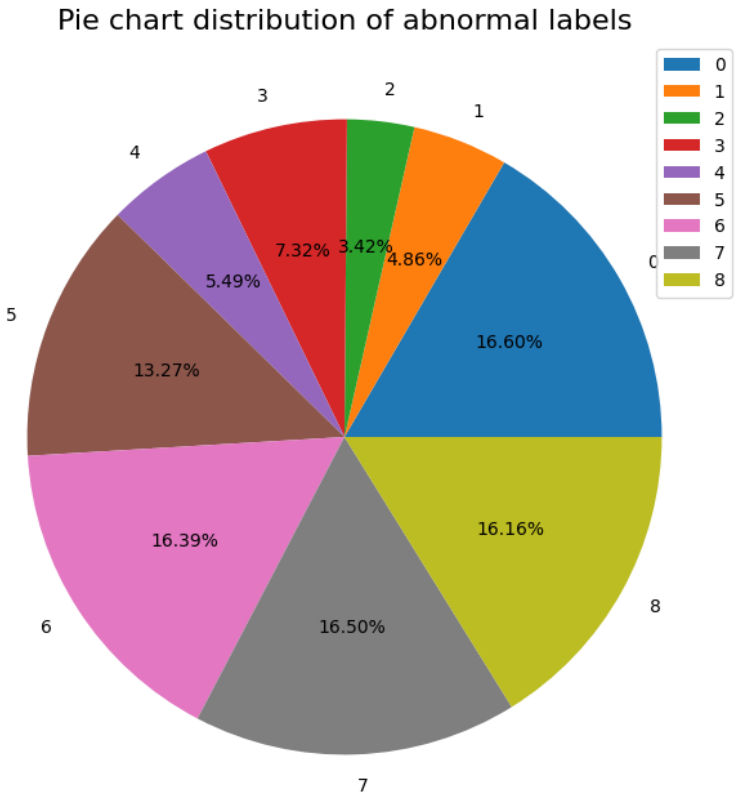
Pie Chart Distribution of normal and abnormal labels in binary classification

3.2.2 Multi-Class Classification

To develop a Multi-Class classification model aimed at identifying kind of the abnormal attack, Therefore, the preprocessing step involved solely removing the normal labeled data from the dataset. the preprocessing step involved data augmentation using Imbalanced-learn (imblearn) library, which is specifically designed to address the issue of class imbalance in machine learning datasets. Class imbalance occurs when the number of instances in each class of a dataset is significantly skewed, which can lead to models being biased towards the majority class and performing poorly on the minority class. Imbalanced-learn provides various tools and techniques to handle class imbalance, including resampling methods such as oversampling (increasing the number of instances in the minority class) and undersampling (reducing the number of instances in the majority class using Tomeklink [38]), as well as algorithms like Synthetic Minority Over-sampling Technique (SMOTE [39]) for generating synthetic samples to balance the classes.By using imblearn, data scientists and machine learning practitioners can improve the performance of models on imbalanced datasets and mitigate the challenges posed by class imbalance in classification tasks.



Pie Chart Distribution of abnormal categories in multi-class classification before data augmentation



Pie Chart Distribution of abnormal categories in multi-class classification after data augmentation

3.2.3 Encoding

For both kinds of classification, the non-numerical features were coded using skit-learn Ordinal Encoder and the numerical values were scaled using skit-learn Standard Scaler

- **Ordinal Encoder:** the Ordinal Encoder in scikit-learn is a preprocessing tool used for converting categorical labels into numerical values in a manner that preserves the ordinality of the categories. This encoder assigns a unique integer to each unique category in a feature, typically starting from zero and incrementing sequentially. The assigned integer values reflect the order or rank of the categories in relation to each other. This transformation allows machine learning algorithms to interpret the categorical data, which is originally in text form, as numerical inputs. The Ordinal Encoder is useful in scenarios where categorical features have intrinsic order or hierarchy, such as "low," "medium," and "high" or "small," "medium," and "large." By encoding these categories with numerical values that respect their inherent order, the model can better understand the relationships between different categories and make more informed predictions based on the ordinal nature of the data.

- **Standard Scaler:** The Standard Scaler in scikit-learn is a preprocessing technique that is utilized for standardizing features by removing the mean and scaling them to unit variance. This transformation is important when working with machine learning algorithms that are sensitive to the scale of input data, such as Support Vector Machines (SVM) or K-Nearest Neighbors (KNN). It works by calculating the mean and standard deviation of each feature in the training data. It then subtracts the mean from each feature and divides by the standard deviation, resulting in a standardized feature where the mean is 0 and the variance is 1. This process ensures that all features are on a similar scale, preventing certain features from dominating the learning process due to their larger magnitudes. By using the Standard Scaler preprocessing method in scikit-learn, machine learning models can perform more efficiently and effectively, especially in scenarios where the input features have different scales or units. This normalization step is crucial for achieving optimal model performance and enhancing the interpretability of the model's results.

3.3 Training and Testing

3.3.1 Modelling

The dataset was split 80% for training and 20% for testing. Also, a random seed was set to ensure reproducibility

3.3.2 Training

In order to reach the best Machine Learning model to predict the label/category, we tried multiple ML models (Decision Tree, Logistic Regression, KNN, SVC and Random Forest). The parameters that affect the model efficiency were changed in each experiment in order to find out the best model

3.3.3 Testing

1. Confusion Matrix: The confusion matrix is a fundamental tool in evaluating the performance of a classification model. It is a table that summarizes the performance of a model by presenting the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions made by the model.
   - True Positive (TP): The instances that are correctly predicted as positive by the model.
   - True Negative (TN): The instances that are correctly predicted as negative by the model.
   - False Positive (FP): The instances that are incorrectly predicted as positive by the model.
   - False Negative (FN): The instances that are incorrectly predicted as negative by the model.

   From the confusion matrix, various metrics can be derived to offer insights into the model's performance, such as accuracy, precision, recall (sensitivity), specificity, and F1 score. These metrics help in understanding how well the model is performing in terms of correctly identifying positive and negative instances.

Confusion Matrix

			Ground Truth Label	
	Total Observations (n)		*has disease*	*no disease*
			Condition Positive (CP)	Condition Negative (CN)
Predicted Label	*test positive*	Test Outcome Positive (TOP)	True Positive (TP)	False Positive (FP)
	test negative	Test Outcome Negative (TON)	False Negative (FN)	True Negative (TN)

Confusion Matrix

2. ROC-AUC (Receiver Operating Characteristic - Area Under the Curve): The ROC curve is a graphical representation of the true positive rate (TPR) against the false positive rate (FPR) for different thresholds used in a binary classification model. The ROC curve visually represents the trade-off between sensitivity (true positive rate) and specificity (true negative rate) across various threshold values. The Area Under the Curve (AUC) of the ROC curve, denoted as ROC-AUC, quantifies the overall performance of the classification model. A higher ROC-AUC score indicates better discrimination ability of the model in distinguishing between positive and negative classes. An ROC-AUC score of 0.5 suggests random prediction performance, while a score of 1.0 represents perfect classification.

Chapter 4: Results & Discussion

4.1 Binary Classification

4.1.1 Decision Tree

NO	Max depth	Train Accuracy	Test Accuracy	Recall	Precision	F1 score
1	5	94%	93%	93%	94%	93%
2	25	99%	95%	95%	95%	95%
3	20	98%	95%	95%	95%	95%
4	15	96%	95%	95%	95%	95%

Table 2 Decision tree Binary Classification EXP

4.1.2 Random Forest

NO	n_estimators	Max depth	Train Accuracy	Test Accuracy	Recall	Precision	F1 score
1	50	5	93%	93%	93%	94%	93%
2	50	25	97%	96%	96%	96%	96%
3	150	15	97%	96%	96%	96%	96%

Table 3 Random Forest Binary Classification EXP

4.1.3 Logistic Regression

NO	Solver	C	Train Accuracy	Test Accuracy	Recall	Precision	F1 score
1	liblnear	5	93%	93%	93%	93%	93%
2	Newton-cholesky	15	93%	93%	93%	94%	93%
3	Newton-cholesky	30	93%	93%	93%	94%	93%

Table 4 Logistic Regression Binary Classification EXP

4.1.4 KNN

NO	N neighbors	Train Accuracy	Test Accuracy	Recall	Precision	F1 score
1	10	94%	94%	94%	94%	94%
2	50	94%	94%	94%	94%	94%
3	20	94%	94%	94%	94%	94%

Table 5 KNN Binary Classification EXP

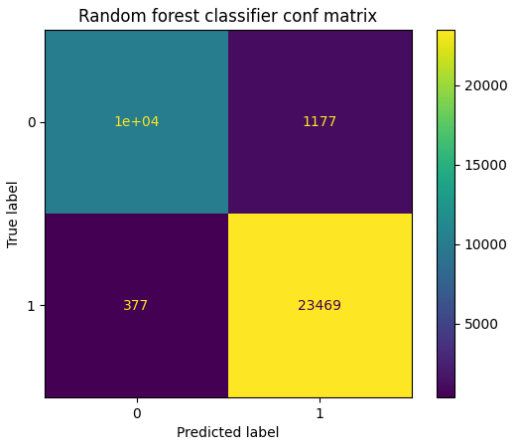
4.1.5 SVC

NO	C	Gamma	Kernel	Train Accuracy	Test Accuracy	Recall	Precision	F1 score
1	5	2.5	rbf	99%	94%	94%	94%	94%

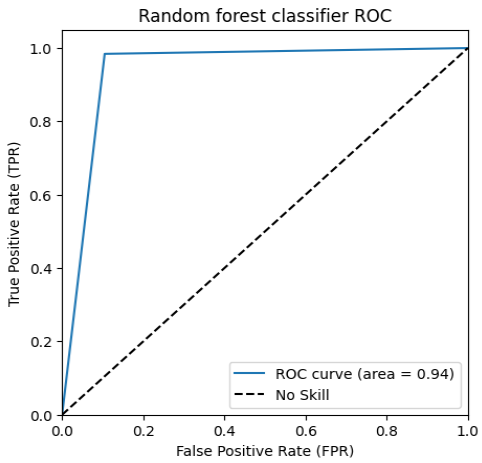
Table 6 SVC Binary Classification EXP

4.1.6 Best Binary Classification Model

The best model is the second Random Forest Classifier experiment which gave the best confusion matrix and roc-auc plot



random forest exp2 conf matrix



random forest exp2 roc-auc

4.2 Multi-Class classification

4.2.1 Without data augmentation

4.2.1.1 Decision Tree

NO	Max depth	Train Accuracy	Test Accuracy	Recall	Precision	F1 score
1	5	77%	77%	77%	79%	73%
2	25	87%	79%	79%	79%	78%

Table 7 Decision tree multi class Classification EXP

4.2.1.2 Random Forest

NO	N_estimeters	Max depth	Train Accuracy	Test Accuracy	Recall	Precision	F1 score
1	50	5	78%	78%	78%	73%	73%
2	50	25	87%	83%	81%	81%	79%

Table 8 random forest multi class Classification EXP

4.2.1.3 Logistic regression

NO	C	Train Accuracy	Test Accuracy	Recall	Precision	F1 score
1	5	74%	74%	74%	71%	69%
2	15	74%	74%	74%	72%	69%

Table 9 logistic regression multi class Classification EXP

4.2.1.4 KNN

NO	N neighbors	Train Accuracy	Test Accuracy	Recall	Precision	F1 score
1	10	75%	76%	76%	74%	74%
2	50	76%	76%	76%	74%	73%

Table 10 KNN multi class classification EXP

4.2.1.5 SVC

NO	C	Gamma	Kernel	Train Accuracy	Test Accuracy	Recall	Precision	F1 score
1	5	2.5	linear	76%	76%	76%	74%	72%

Table 11 SVC multi class classification EXP

4.2.2 With data augmentation

4.2.2.1 Decision Tree

NO	Max depth	Train Accuracy	Test Accuracy	Recall	Precision	F1 score
1	5	77%	77%	77%	77%	76%
2	25	99%	96%	96%	96%	96%
3	20	97%	95%	95%	95%	95%

Table 12 AUG Decision tree multi class classification EXP

4.2.2.2 Random Forest

NO	N_estimeters	Max depth	Train Accuracy	Test Accuracy	Recall	Precision	F1 score
1	50	5	83%	83%	83%	82%	81%
2	50	15	95%	94%	94%	94%	94%
3	50	25	99%	97%	97%	97%	97%
4	50	20	98%	96%	96%	96%	96%

Table 13 AUG Random Forest multi class classification EXP

4.2.2.3 Logistic Regression

NO	C	Train Accuracy	Test Accuracy	Recall	Precision	F1 score
1	5	71%	71%	71%	70%	69%
2	15	70%	70%	70%	69%	68%
3	30	70%	70%	70%	69%	68%

Table 14 AUG Logistic Regression multi class classification EXP

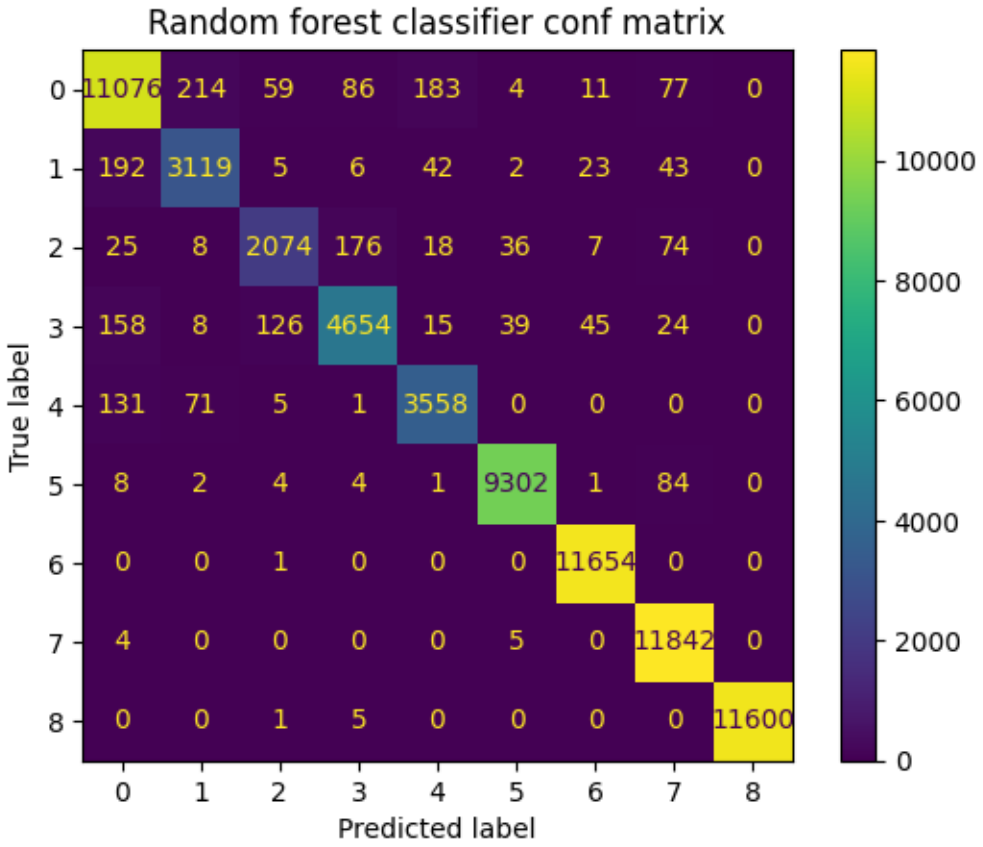
4.2.2.4 KNN

NO	N neighbors	Train Accuracy	Test Accuracy	Recall	Precision	F1 score
1	15	92%	92%	92%	92%	92%
2	50	86%	86%	86%	86%	86%
3	75	84%	84%	84%	84%	84%

Table 15 AUG KNN multi class classification EXP

4.2.2.5 Best Multi-class Classification Model

The best model is the Third Random Forest Classifier experiment which gave the best confusion matrix



Best model- Random Forest multi class classification Classifier conf mat

References

[1] Seemma, P. S., Nandhini, S., & Sowmiya, M. (2018). Overview of cyber security. International Journal of Advanced Research in Computer and Communication Engineering, 7(11), 125-128.

[2] Yaacoub, J. P. A., Salman, O., Noura, H. N., Kaaniche, N., Chehab, A., & Malli, M. (2020). Cyber-physical systems security: Limitations, issues and future trends. Microprocessors and microsystems, 77, 103201.

[3] Cremer, F., Sheehan, B., Fortmann, M., Kia, A. N., Mullins, M., Murphy, F., & Materne, S. (2022). Cyber risk and cybersecurity: a systematic review of data availability. The Geneva Papers on risk and insurance-Issues and practice, 47(3), 698-736.

[4] Ghelani, D. (2022). Cyber security, cyber threats, implications and future perspectives: A Review. Authorea Preprints.

[5] Sonkor, M. S., & García de Soto, B. (2021). Operational technology on construction sites: A review from the cybersecurity perspective. Journal of Construction Engineering and Management, 147(12), 04021172.

[6] Li, Y., & Liu, Q. (2021). A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. Energy Reports, 7, 8176-8186.

[7] Glăvan, D., Răcuciu, C., Moinescu, R., & Eftimie, S. (2020). Sniffing attacks on computer networks. Scientific Bulletin" Mircea cel Batran" Naval Academy, 23(1), 202A-207.

[8] Chen, M., & Yan, M. (2023). How to protect smart and autonomous vehicles from stealth viruses and worms. ISA transactions.

[9] Albalawi, A. M., & Almaiah, M. A. (2022). Assessing and reviewing of cyber-security threats, attacks, mitigation techniques in IoT environment. J. Theor. Appl. Inf. Technol, 100, 2988-3011.

[10] Molotkov, S. N. (2020). Trojan horse attacks, decoy state method, and side channels of information leakage in quantum cryptography. Journal of Experimental and Theoretical Physics, 130, 809-832.

[11] Popoola, S. I., Adebisi, B., Hammoudeh, M., Gui, G., & Gacanin, H. (2020). Hybrid deep learning for botnet attack detection in the internet-of-things networks. IEEE Internet of Things Journal, 8(6), 4944- 4956.

[12] Sun, X., Dai, J., Liu, P., Singhal, A., & Yen, J. (2018). Using Bayesian networks for probabilistic identification of zero-day attack paths. IEEE Transactions on Information Forensics and Security, 13(10), 2506-2521.

[13] Ali, S., Rehman, S. U., Imran, A., Adeem, G., Iqbal, Z., & Kim, K. I. (2022). Comparative Evaluation of AI- Based Techniques for Zero-Day Attacks Detection. Electronics, 11(23), 3934.

[14] Guo, Y. (2022). A review of Machine Learning-based zero-day attack detection: Challenges and future directions. Computer Communications.

[15] Zoppi, T., Ceccarelli, A., & Bondavalli, A. (2021). Unsupervised algorithms to detect zero-day attacks: Strategy and application. Ieee Access, 9, 90603-90615.

[16] Kumar, V., & Sinha, D. (2021). A robust intelligent zero-day cyber-attack detection technique. Complex & Intelligent Systems, 7(5), 2211-2234.

[17] Rashid, A. B., Ahmed, M., Sikos, L. F., & Haskell-Dowland, P. (2022). Anomaly detection in cybersecurity datasets via cooperative co-evolution-based feature selection. ACM Transactions on Management Information Systems (TMIS), 13(3), 1-39.

[18] Alabadi, M., & Celik, Y. (2020, June). Anomaly detection for cyber-security based on convolution neural network: A survey. In 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) (pp. 1-14). IEEE.

[19] Rajbahadur, G. K., Malton, A. J., Walenstein, A., & Hassan, A. E. (2018, June). A survey of anomaly detection for connected vehicle cybersecurity and safety. In 2018 IEEE Intelligent Vehicles Symposium (IV) (pp. 421-426). IEEE.

[20] Ravinder, M., & Kulkarni, V. (2023, January). A Review on Cyber Security and Anomaly Detection Perspectives of Smart Grid. In 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 692-697). IEEE.

[21] Kuzlu, M., Fair, C., & Guler, O. (2021). Role of artificial intelligence in the Internet of Things (IoT) cybersecurity. Discover Internet of things, 1, 1-14.

[22] Geluvaraj, B., Satwik, P. M., & Ashok Kumar, T. A. (2019). The future of cybersecurity: Major role of artificial intelligence, machine learning, and deep learning in cyberspace. In International Conference on Computer Networks and Communication Technologies: ICCNCT 2018 (pp. 739-747). Springer Singapore.

[23] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. CRC press.

[24] Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal

[25] Breiman, L. (2001). Random forests. Machine learning

[26] Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest

[27] Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression. John Wiley & Sons.

[28] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

[29] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media.

[30] Cover, T., & Hart, P. (1967). Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory, 13(1), 21-27.

[31] Cortes, Corinna, and Vapnik, Vladimir. "Support-vector networks." Machine learning 20.3 (1995): 273-297.

[32] Scholkopf, Bernhard, et al. "Comparing support vector machines with Gaussian kernels to radial basis function classifiers." IEEE transactions on signal processing 45.11 (1997): 2758-2765.

[33] Burges, Christopher JC. "A tutorial on support vector machines for pattern recognition." Data mining and knowledge discovery 2.2 (1998): 121-167.

[34] Guha Roy, D., & Srirama, S. N. (2021). A blockchain-based cyber-attack detection scheme for decentralized Internet of Things using software-defined network. *Software: practice and experience*, *51*(7), 1540-1556.

[35] Tan, L., Yu, K., Ming, F., Cheng, X., & Srivastava, G. (2021). Secure and resilient artificial intelligence of things: a HoneyNet approach for threat detection and situational awareness. *IEEE Consumer Electronics Magazine*, *11*(3), 69-78.

[36] Matin, I. M. M., & Rahardjo, B. (2019, November). Malware detection using honeypot and machine learning. In 2019 7th international conference on cyber and IT service management (CITSM) (Vol. 7, pp. 1-4). IEEE.

[37] Naik, N., & Jenkins, P. (2018, July). A fuzzy approach for detecting and defending against spoofing attacks on low interaction honeypots. In *2018 21st International Conference on Information Fusion (Fusion)* (pp. 904-910). IEEE.

[38] I.Tomek, “Two modifications of CNN,” In Systems, Man, and Cybernetics, IEEE Transactions on, vol. 6, pp 769-772, 1976.

[39] N. V. Chawla, K. W. Bowyer, L. O.Hall, W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” Journal of artificial intelligence research, 321-357, 2002.