USING ARTIFICIAL INTELLIGENCE FOR AUTOMATED INCIDENCE RESPONSE IN CYBERSECURITY

Uzoma, Joseph[1]; Falana, Olakunle[2]; Obunadike, Callistus[2]; Oloyede, Kunle[3]
Obunadike, Echezona[4]
[1,2,3,4] Department of Computer Science, Austin Peay State University,
Clarksville USA.

ABSTRACT

This paper delves into the critical evaluation of four machine learning algorithms: Random Forest Classifier, Support Vector Machine (SVM), Decision Trees, and KModes. These algorithms are analyzed using key metrics such as accuracy score, precision score, F1 score, and recall score. Two data sizes, size one and size two, were employed for training and testing, featuring varying numbers of anomalous and normal HTTP requests. The performance of each algorithm is illustrated through line graphs, providing a clear comparison between the two data sizes. For the first data size (10 anomalous and 20 normal HTTP requests), the Decision Tree classifier demonstrated the highest accuracy score at 0.9, followed by SVM (0.85), Random Forest Classifier (0.83), and KModes (0.6). Precision scores were relatively consistent, with SVM, Decision Tree, and KModes around 0.67, and Random Forest Classifier slightly lower at 0.65. Similarly, recall and F1 scores displayed a similar trend. In the second data size (100 anomalous and 200 normal HTTP requests), all four algorithms achieved an accuracy score of 0.98. Precision, recall, and F1 scores were approximately 0.67 for all four algorithms. This study aims to enhance incident detection skills by exploring the potential of machine learning algorithms in detecting malicious HTTP requests. It demonstrates the potential of AI in cybersecurity, emphasizing the ability to distinguish normal requests from malicious ones. In conclusion, the Decision Tree Classifier excelled in the first data size, while all four algorithms demonstrated strong performance in the second data size, except for KModes, which consistently exhibited lower accuracy and other metric scores. This research provides insights into how AI and machine learning can bolster cybersecurity efforts, particularly in the context of malicious HTTP request detection.

Keywords:

Cybersecurity, Malicious HTTP request detection, Random Forest Classifier, Support Vector Machine, Decision Trees, and KMode.

1. INTRODUCTION

AI can help automate incident response in cybersecurity, which will shorten the time it takes to identify and mitigate hazards. AI can also examine enormous volumes of data, looking for trends and abnormalities that can help forecast and stop cyberattacks. To provide the highest level of security, AI should be used in conjunction with other cybersecurity strategies. The purpose of this study is to assess the advantages and challenges of using AI to automate incident response in cybersecurity.

1.1 Statement of the Problem

Cybercrime is an emerging worry, and it is essential to take precautions to protect ourselves from it, according to Jain, Kant, and Varshney (2019). Using firewall software, antivirus software, internet security software, strong passwords, and making sure the operating system is up to date are just a few of the methods available to achieve this. Security teams are using tools like machine learning, natural language processing, and deep learning in conjunction with security logs to automate repetitive processes, speed up threat detection and response, and improve the accuracy of their operations. However, there are concerns regarding the dependability and credibility of AI-based incident response systems, in addition to ethical issues like bias and privacy. The primary challenge addressed by AI for Automated Incident Response in Cybersecurity is the escalating complexity and sophistication of cyber-attacks.

1.2 Background of the Study

The use of artificial intelligence (AI) to automate many parts of incident response in cybersecurity has been studied. Insights on the usage of AI for automated incident response in cybersecurity are provided by a paper by Mohit Jain, Kshama Kant, and Anupam Varshney (2019), which also emphasizes how AI has the ability to transform incident response procedures. Artificial intelligence (AI) tools like machine learning, natural language processing, and deep learning offer the ability to automate many incident response processes, increasing the responsiveness and precision of efforts. This article offers a thorough analysis of how AI is used in cybersecurity for automated incident response.

1.3 Purpose of the Study

Researchers have looked into employing artificial intelligence (AI) to automate incident response in cybersecurity. A paper by Mohit Jain, Kshama Kant, and Anupam Varshney from 2019 explores the use of AI for automating incident response in cybersecurity and highlights how AI has the potential to change incident response procedures. Automating numerous incident response parts with the use of technologies like machine learning, natural language processing, and deep learning in artificial intelligence could improve the effectiveness and speed of response operations. This paper provides a comprehensive review of the application of AI to cybersecurity incident response automation.

1.4 Research Questions

The focus of this research will be on the following research inquiries:

- How can we ensure that the automated incident response dataset and models are accurately efficient?
- What are the outcomes of training different machine learning algorithms in terms of performance?

1.5 Significance to the Study

Artificial intelligence can help improve incident response in cybersecurity by automating several aspects of the process, such as detecting and analyzing potential threats, prioritizing alerts, and taking immediate action. It can also improve the speed and accuracy of response efforts and reduce the workload on human security teams (Frost & Sullivan, 2019).

1.6 Limitation of the Study

The study conducted by Mohit Jain, Kshama Kant, and Anupam Varshney (2019) concentrates on the potential advantages of utilizing AI for automating incident response in cybersecurity but overlooks the constraints and difficulties associated with implementing such systems. To use ML algorithms, a large amount of data and effective hardware resources are needed. In general, ML models are built and trained to recognize cyberattacks. A model might not work well in detecting different assaults or guarding against changing cyber-attacks. Finding previously unseen behaviors to detect can be difficult, and these activities may differ greatly from their predecessors (Kamran et al., 2020). Because models are typically trained using historical features from a dataset, the most recent attacks may escape the classifiers, resulting in a lower detection rate and a higher number of false positives. The datasets used for model training and model evaluation are another drawback of machine learning. Most publicly available datasets do not reflect the most current attacks. Despite the existence of anonymization techniques, restrictions and privacy worries prevent the release of data that might be useful for ML. Data in cybersecurity generally comes from a diverse range of heterogeneous log sources, and this heterogeneity might provide challenges for ML models.

1.7 Definition of Terms

- **Incidence response:** Incident response (IR) for an organization is the process of quickly identifying a cyber-incident, minimizing its consequences, containing the damage, and addressing the root cause to reduce the likelihood of repeat events.
- **Machine Learning:** Artificial intelligence (AI) includes machine learning, a subset that enables computer systems to learn and develop naturally from experience without being explicitly programmed.
- **Neural Networks -** a group of algorithms that are intended to recognize patterns and are loosely fashioned after the human brain.

- **Computer security, cybersecurity (cyber security), or information technology security (IT security):** is the defense of computer systems and networks from hostile actors who aim to disrupt or misdirect the services they offer, reveal confidential information inadvertently, steal hardware, steal software, or harm data.
- **Artificial intelligence (AI):** Machine intelligence, as opposed to non-human animal and human intelligence, is the ability to perceive, synthesize, and infer information. Speech recognition, computer vision, interlanguage translation, and various mappings of inputs are a few examples of activities where this is done.
- **Deep learning (DL):** Deep learning is a branch of machine learning that uses neural networks to make decisions and automatically learn from the past. Contrary to traditional machine learning, deep learning algorithms can self-tune and self-improve, negating the need for explicit programming. The focus of this session will be deep learning, a subfield of machine learning that is explored in relation to cybersecurity.
- **Automated Incident Response:** Automated incident response is the use of automated systems and processes to monitor security warnings and react to them using incident response protocols that have been set and taken from an organization's incident response plan. Instead of getting mired down in tedious tasks, this method frees security operations center (SOC) analysts to focus on more strategic and proactive threat hunting efforts.
- **HTTP**: Hyper Text Transfer Protocol. It is the foundation of the World Wide Web and is used to load web pages using hypertext links.
- **HTTP Server**: is a hardware or software application that implements the server component of the HTTP/HTTPS network protocols to function as a server in a client-server situation. It is also referred to as a "web server."
- Server: Is a program or tool that provides a service to the user of another program.

2. LITERATURE REVIEW

As internet technology and systems continue to advance rapidly, cybercrimes and attacks are also on the rise. To combat these threats, AI-based techniques are needed in cybersecurity systems to improve cyberspace security more effectively. Preliminary research on the use of AI for automated incident response in cybersecurity suggests that AI has the potential to significantly improve incident response times and accuracy. By analyzing large amounts of data and detecting anomalies, AI systems can quickly identify and respond to security incidents. Some studies have also found that AI-based incident response solutions can reduce the workload of security analysts while improving the efficiency of incident response operations. However, there are questions about the accuracy and dependability of AI-based incident response systems, as well as ethical issues such as bias and privacy. Some experts think that AI should not replace human decision-making in crisis response, but rather augment and support it (Asaro, et al., 2020).

Overall, the literature suggests that AI can have a significant positive impact on incident response in cybersecurity, but it is crucial to thoroughly evaluate the challenges and limitations involved. Further research is necessary to comprehensively comprehend the effects of AI on incident response and to establish effective strategies for its implementation.

In recent years, there has been significant research on the use of Artificial Intelligence (AI) in incident response for cybersecurity. Different authors have conducted several studies on this topic, such as (Donepudi,2021) who critically investigated the application of AI in the automation industry, Erik and Una-May (2021) who explored using a collated cybersecurity dataset for Machine Learning and AI, and Jari (2022) who conducted a study on AI in cybersecurity. Automated incident response can address various challenges in incident response, including shorter response times and the need for more consistent and successful response activities. A poll suggests that 70% of firms expect a shortage of cybersecurity workers in 2020. By automating routine tasks like threat identification, containment, and remediation, organizations can reduce the risk of human error and improve the overall effectiveness of incident response.

Artificial intelligence (AI) has the potential to revolutionize cybersecurity by automating incident response. By analyzing massive volumes of data in real-time, AI algorithms can rapidly detect potential security threats and take swift and effective action as needed. This can help businesses improve their defenses against intrusions and significantly enhance the speed and accuracy of incident response. One specific area where AI can be of great use is the analysis of security logs, reducing the workload on security personnel and freeing up resources for more complex tasks. However, it is crucial that the technology is deployed ethically and transparently, and the use of AI in incident response requires careful consideration of ethical and privacy issues.

Artificial intelligence (AI) can make a significant impact on the cybersecurity industry. By automating the process of detecting and responding to cyber threats, AI can reduce the time required to identify and address risks. Moreover, it can analyze vast amounts of data to detect patterns and anomalies that can be used to predict and prevent cyberattacks. AI-based systems can even learn from previous threats and adapt to new ones, which may enhance their effectiveness over time. However, it is important to note that AI is not a one-size-fits-all solution and should be used in conjunction with other cybersecurity measures to ensure optimal protection. Sarker et al. (2021) proposed a generic definition of cybersecurity. Information security, network security, operational security, application security, Internet of Things (IoT) security, cloud security, and infrastructure security are all considered to be part of cybersecurity, according to Sarker et al. (2021). Its goal is to protect sensitive and confidential information from cyberattacks and cybercriminals, such as personal, governmental, and industrial information. Morgan (2019) predicts that investment in cybersecurity worldwide will increase by more than $1 trillion from 2016 to 2021. The widespread use of the internet and the constant exchange of vast amounts of data have made cybersecurity an urgent matter. The frequency and potency of cyberattacks are increasing at an alarming rate, with cybercriminals constantly improving their tailored attacks while reducing the cost (Stevens, 2018). Traditional cybersecurity measures, Ineffective against the inventive and developing ways of cyberattacks are technologies like network protection systems and computer security systems (Kabbas et al., 2020; Truong et al., 2020). Artificial intelligence (AI) is one of the solutions that may be employed effectively in cybersecurity, therefore new ways are required to tackle rising cyber threats and malware (Soni, 2020). Two recently created AI domains, machine learning (ML) and deep learning (DL), have proven successful in thwarting cyberattacks (Zeadally et al., 2020).

As a significant branch of computer science, AI deals with the development of intelligent and independent systems that mimic the human brain (Helm et al., 2020). In today's digital age, cybersecurity is a significant concern, and AI's role in cybersecurity is more important than ever. In addition to cybersecurity, AI has recently emerged as a major participant in the fields of natural language processing, gaming, healthcare, manufacturing, and education (Zeadally et al., 2020). In contrast to previous systems, AI has strong data analytics skills that enable it to analyze enormous amounts of electronic data quickly, effectively, and precisely (Truong et al., 2020). The AI system is a useful tool for cybersecurity because it can predict upcoming cyberattacks based on past threats, even if those threats change (Zeadally et al., 2020). In the current digital era, Security Operation Center (SOC) analysts confront difficulties in efficiently managing the huge volume, speed, and variety of data across different security devices including firewalls, IDS, and SIEM. These tools lack integration, have inconsistent workflows, lack standardization for data transmission, and operate autonomously. As a result, SOC analysts struggle to have a complete picture of their organization's security posture and must continuously involve humans in the security incident response process. The shortage of security experts adds to the complexity of the situation, and the growing threat landscape makes it even more challenging to manage security effectively. (Morgan, 2019).

The advancement of technology in automation and autonomy has become increasingly important in the fight against cyber threats. It plays a crucial role in detecting, mitigating, and preventing such threats. However, obtaining useful cyber threat intelligence (CTI) for automated processes remains a challenge. As such, the use of machine intelligence has become essential in both defensive and offensive cybersecurity operations to enable automation and autonomous/semi-autonomous systems. This underscores the significance of developing and maintaining technological solutions for defensive cyber operations. AI/ML-based cybersecurity systems are becoming increasingly important in the face of growing threats and their complexity. They enable fast and automated responses to cyber threats by analyzing large sets of data and identifying suspicious patterns in real-time. Such systems can also prevent attacks on a large scale by providing automatic software updates based on sophisticated AI/ML analysis. AI approaches are widely used by email providers to block unwanted photos, identify phishing, malware, and fraudulent payments. Other service providers also employ ANN-based models to recognize and categorize phishing and virus emails. Because it doesn't rely on static signatures like traditional anti-virus systems do, AI/ML is particularly helpful for malware detection and anti-virus defense. Intrusion detection, phishing and spam detection, threat identification and characterization, and user behavioral analytics are the key uses for AI/ML in cybersecurity.

2.1 Related Empirical of Literature

In their research titled "Predicting Cyber Security Incidents using Machine Learning Algorithms: A Case Study of Korean SMEs," Mohasseb et al (2019) analyze a dataset collected from five small and medium-sized companies in South Korea. The dataset consists of cyber security incidents and the response actions taken. The study aims to investigate how data collected from multiple companies representing different incidents can help in improving classification accuracy and assist classifiers in distinguishing between different types of incidents. To achieve this, the authors develop a model using text mining methods, such as n-gram, bag-of-words, and machine learning algorithms, for the classification of incidents and their response actions. Experimental results show that the classifiers perform well in predicting different types of responses and malware. Although AI-based automated incident response systems are superior to conventional methods in cybersecurity, their performance measures such as False Positive Rate (FPR), Detection Rate (DR), and Mean Time to Detect (MTTD) may vary depending on the implementation, dataset, and threat landscape. Therefore, human oversight and intervention are still necessary in many circumstances, and AI-based systems should not be solely relied upon for incident response (as stated in "Cylance: AI-Driven Incident Response"). In the study by Fan et al. (2016), a method for detecting malicious sequential patterns was proposed. This was achieved by mining instruction sequences from a set of file samples and using these patterns to construct a Nearest-Neighbor (ANN) classifier for malware detection. The data mining framework consisted of the proposed sequential pattern mining method and the ANN classifier. Wang et al. (2006) proposed an integrated architecture for defending against surveillance spyware by using features extracted from both static and dynamic analysis. The features were ranked based on their information gains, and a machine learning algorithm was utilized.

Abou-Assaleh et al. (2004) presented a method for detecting malicious code using Common N-Gram analysis (CNG), which is based on byte n-gram analysis and relies on profiles for class representation. In the study by Shabtai et al. (2012), OpCode n-gram patterns were extracted from disassembled files to detect unknown malicious code. These patterns were used as features for the classification process. Several studies have employed machine learning techniques for detecting and categorizing malicious code. For instance, in (Hou et al., 2010), a method was proposed for identifying malicious web pages by analyzing their characteristics through machine learning techniques. Similarly, in (Zhang et al., 2007), the authors proposed an approach for automatically detecting malicious code through n-gram analysis, with selected features based on information gain. Furthermore, in (Elovici et al., 2007), the authors developed a machine learning-based approach for detecting malicious code in suspicious executable files using three algorithms: Decision trees, Neural Networks, and Bayesian Networks.

Arpitha et al. (2021) conducted research on the use of machine learning techniques for detecting and notifying cyber-attacks. With the increasing prevalence of cybercrime, it has become important for ethical hackers to identify vulnerabilities and recommend mitigation strategies. Machine learning is a promising solution to address cyber security challenges, helping identify threats more efficiently and reduce security analysts' workload. Adaptive machine learning methods can lead to higher detection rates, lower false alarm rates, and reasonable computation and communication costs. However, the task of identifying cyber-attacks is more challenging than other machine learning applications, posing difficulties for the intrusion detection community to effectively employ machine learning. Joseph (2019) conducted a study on the potential risks associated with artificial intelligence in cyber security, and the role of humans in mitigating these risks. The paper examines reported failures of AI systems and predicts that both the frequency and severity of such failures will increase in the future. The author suggests that AI safety can be improved by adopting ideas developed by cybersecurity experts. While the safety failures of narrow AIs are at a moderate level of criticality similar to those in cybersecurity, failures of general AIs can have a catastrophic impact. AI-powered automated incident response can improve the efficiency and effectiveness of cybersecurity operations, enabling businesses to protect themselves against increasingly sophisticated cyber threats. Apruzzese et al (2022) conducted a study on the application of machine learning (ML) in cybersecurity. The paper provides a comprehensive overview of the benefits of using ML in cybersecurity, including its ability to augment human-driven detection methods and address additional tasks. Collaboration between stakeholders is necessary to overcome challenges and further progress in the field of ML in cybersecurity. Goni, Ibrahim, et al (2020) conducted research on the Machine Learning Approach to Cybersecurity and Cyber Forensics. With the widespread adoption of cloud computing and the internet of things, nations worldwide have become increasingly connected through global networks. This study examines the use of machine learning techniques in cybersecurity and cyber forensics, finding that cybersecurity relies on maintaining data confidentiality, integrity, and validity. The authors also identified ten critical steps in achieving cybersecurity, which include network security, user education and awareness, malware prevention, removable media control, secure configuration, user privilege management, incident management, monitoring, and remote and mobile working. The research results provide insights for future investigations on the utilization of deep learning, computational intelligence, and soft computing in the fields of cybersecurity and cyber forensics.

2.2 Security Expert Systems

A form of computer program known as an expert system helps a human expert make choices. Its knowledge base and inference engine combine to provide security rules that form the cornerstone of the system's judgments (Tyugu, 2016). Many industries, including medicine, finance, and cybersecurity, use expert systems extensively. They can be little or big, intricate hybrid systems that handle complex concerns and issues. The domain knowledge and operational knowledge of security decision rules are described in the knowledge base phase of the cybersecurity expert system structure. The next step is the inference engine phase, which draws conclusions from the knowledge base and infers brand-new facts. Depending on how the reasoning is carried out, expert systems can be used to solve a variety of problems. A solution is produced in the case-based reasoning (CBR) technique, for instance, by applying the prior answer to a new problem situation. A specific problem is solved by remembering analogous earlier problems. Rule-based systems (RBS), which use rules established by experts to solve problems, are another method of problem-solving. The condition component and the action component make up an RBS. The problem is assessed in the condition part, and the analysis is used to decide what should be done. Guidelines and regulations are also used by cybersecurity expert systems to thwart cyberattacks. For instance, the security system deems a process secure if it has been assessed against the knowledge base and is known to be so. The system declares the process to be a threat and stops it if it is unknown. If the knowledge base does not have information on the process, the inference engine uses sets of rules to determine the machine's state, which can be severe, moderate, or safe. Based on the machine's state, the system notifies the user or manager of the machine's status, and the inference is detected by the knowledge base. Utilizing a rule-based model in cybersecurity can enable an expert system to make decisions like a human security expert and apply intelligent reasoning to solve intricate cybersecurity problems. Therefore, the integration of a cybersecurity expert system model, due to its computing abilities and decision-making intelligence, could be an advantageous aspect of AI-based cybersecurity.

2.3 Benefits of AI in Cybersecurity

Organizations that have implemented AI techniques in their cybersecurity operations have reaped significant benefits, as evidenced by increased ROI (Lazic, 2019). One example is the Siemens Cyber Defense Center, which uses modern technologies such as AI to monitor and protect Siemens' global IT and operational technology infrastructure. With the help of AI, security analysts can quickly and accurately detect threats, automate repetitive security operations, and provide comprehensive threat intelligence to Siemens' IT and OT teams. This knowledge can then be used to proactively enhance Siemens' security posture, reduce the risk of cyber-attacks, and mitigate their impact. The system was able to handle up to 60,000 attacks per unit of time with fewer than 12 staff members while maintaining good system performance. AI-based cybersecurity can identify new threats by analyzing past threat patterns using machine learning methods, deep learning techniques, and natural language processing. Machine learning algorithms can analyze vast amounts of security data to identify potential threats. Deep learning techniques can analyze real-time data from sensors and devices to identify potential risks, while natural language processing can analyze text-based data to detect potential threats or attacks. The use of AI in cybersecurity offers a more efficient and cost-effective approach to identifying and responding to threats. In fact, it has been found that AI can reduce costs by up to 12% while also providing better threat detection and response capabilities. With the shift towards automated algorithm mitigation in cybersecurity, AI is becoming increasingly important in solving complex security issues. Unlike traditional methods that rely on known threats and may miss unusual intrusion activities, AI can detect new and sophisticated attack patterns. For instance, AI can monitor privileged internet activities and identify any changes in privileged access as potential threats. By using predictive methods, AI gives security teams an advantage in proactively stopping attacks before they cause damage.

A UK-based startup called Darktrace uses machine learning to spot dangers and patterns across a range of sectors, including retail, manufacturing, energy, and transportation. AI-based methods can manage the vast volume of data and enhance network security. AI-based autonomous detection and response to threats have helped to lighten the load on security teams as they deal with an enormous volume of security issues. For security professionals, the daily generation and transfer of enormous amounts of security data can be challenging. AI can assist with a more thorough analysis of suspicious processes and actions. Replacing manual methods, which can be time-consuming when responding to novel situations, AI-based systems continue to learn over time and respond better to threats and attacks. AI can identify attacks based on the characteristics of the application and overall network activity. Over time, AI memorizes the normal traffic status and sets a limit for normal activities, so that any abnormal deviation is marked as an attack.

2.4 Key challenges in AI-driven Cybersecurity Applications

Due to the necessity for massive memory and processing power, developing an Artificial Intelligence system necessitates processing enormous amounts of data and input samples, which consumes a lot of time and resources. Additionally, costly, and sophisticated resources are needed for the implementation of AI technology. False alarms are a common event that end users must deal with on a regular basis since they can disrupt the entire business environment and interfere with critical answers. To decrease false alarms and keep the security level, fine-tuning is employed as a tradeoff process. AI-based systems can be targeted and attacked using a variety of techniques, including data poisoning, model theft, and adversarial inputs. An AI model consists of four essential components, including data perception, learning, decision-making, and action-taking. These components operate in a highly complex environment, where each component must interact and have mutual dependencies. A wrong decision can result from an incorrect perception, and each component is exposed to different attacks and threats. For example, decisions can be vulnerable to traditional cyber-attacks, while perception is susceptible to training attacks.

Conclusion: The idea of consistency is illogical because maintaining a measure of uncertainty requires connecting the pieces that prevent the system from malfunctioning. It is essential to have a successful strategy for individually validating the judgments, logic precision, and risk assessments for various AI and ML components. New methods must be put into practice in order to support systems' requirements and respond to varied attacks. However, the use of AI in cybersecurity may bring about fresh dangers, endangering online safety. Ai’s ability to consistently detect and prevent cyber-attacks has also led to an increase in more complex and sophisticated attacks by motivated attackers who have access to AI techniques at a lower cost. This has resulted in a rise in cybercrime rates, as reported in "The Economic Impact of Cybercrime" by CSIS in February 2018. To counter these threats, organizations must emphasize the human element in AI-based cybersecurity solutions. This includes training employees to recognize and respond to potential threats, implementing effective cybersecurity risk management policies and procedures, and continuously evaluating and improving their defenses. The risk of complacency also poses a challenge, as individuals may become overly dependent on automation and overlook crucial aspects of cybersecurity that require human attention. However, the dangers of the human component of complacency in AI-based cybersecurity solutions are not sufficiently covered. A fundamental barrier in using AI to solve real-world cybersecurity issues is the collecting, administration, and processing of unquantified data (structured, semi-structured, unstructured, or meta-data).

2.5 Model Training and Algorithms Used

To make a learning algorithm capable of learning, training models include feeding it training data. The training data must have the target attribute, which contains the right response. The target property in this study is the Categorical variable. To create a Machine Learning model that captures these patterns, the learning algorithm examines the training data to look for patterns that link the qualities of the input data with the target attribute (variable). The generated model can then be used to make predictions on new data when the target attribute is unknown. Several techniques, including Nave Bayes, Logistic Regression, Support Vector Machine, Decision Tree, K-Nearest Neighbor, and Random Forest, were used to train the dataset for our models. In addition, we applied. We trained the dataset for our models using a variety of methods, including Nave Bayes, Logistic Regression, Support Vector Machine, Decision Tree, K-Nearest Neighbor, and Random Forest. The KModes Clustering Algorithm was also used to group data with related attributes. According to Olufemi et al. (2023) "*Logistic regression is a machine learning classification technique used for analyzing datasets with one or more independent (PIE) variables to predict the outcome. It is also a statistical method employed to determine the result based on the input variables*". Obunadike et al (2023) stated that the logistic regression model assumes a linear relationship between the independent variables (PIE) and the target variable (DORT).

- **KModes Clustering Algorithm**

The KModes Clustering Algorithm is an unsupervised learning algorithm that classifies data points into categories based on similarities and differences. It clusters categorical data using Modes rather than means and employs the Elbow curve to determine the ideal K value. KModes clustering is particularly suitable for categorical variables compared to other clustering algorithms because it accounts for the number of matches and mismatches between categories, KModes use a matching dissimilarity metric. In contrast to previous clustering techniques, KModes allows for both binary and multi-categorical variables in the same dataset.

- **Random Forest Algorithm (RF)**

The Random Forest algorithm works by randomly selecting k features and forming a decision tree, then repeating this n times, forecasting outcomes with a random variable, and guessing which category the new record belongs to. It may be tweaked using Python's scikit learn package.

- **Decision Tree Algorithm (CART)**

Using if-then-else decision rules, the Decision Tree Algorithm divides a dataset into smaller subgroups, choosing the most crucial traits, and producing predictions.

- **Support Vector Machine Algorithm (SVM)**

The kernel trick is used by the Support Vector Machine to determine an ideal border between outputs when dealing with non-linear connections. It splits new items into two distinct groups.

- **Logistic Regression (LR)**

Logistic regression and linear regression both consider the label's category and value, respectively.

- **Naive Bayes (NB)**

Naive Bayes is a classification technique that makes predictions without using real-world data since it assumes features are independent and have no correlation.

- **Linear Discriminant Analysis (LDA)**

A response variable is divided into two or more classes using the machine learning approach of LDA, and predictions are based on the likelihood that a fresh input data set belongs to each class.

3. METHODS

This chapter explains the method used to examine the usage of Artificial Intelligence (AI) for automated incident response in cybersecurity. The chapter also describes the types of data used, method of collection, model specification, model estimation as well as data presentation and analysis techniques plus the implementation of the model obtained.

3.1 Model Evaluation

The following performance measures were used to rate the models: Accuracy: This metric measures the proportion of accurate predictions to all input samples. In other words, it represents the percentage of all accurate predictions.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+T}$$

Precision is calculated by dividing the total number of correctly positive results by the total number of positive results that the classifier anticipated.

$$Precision = \frac{TP}{TP+FP}$$

F1-Score: F1-Score is used to gauge how accurate a test is. The Harmonic Mean of memory and precision is the F1 Score. The F1 Score has a range of [0, 1]. It informs you of your classifier's precision (the proportion of instances it properly classifies) and robustness (the proportion of instances it does not miss).

$$F\text{-}measure = \frac{2*Recall*Precision}{Recall+Preci}$$

Remember that this ratio is the total number of relevant samples (all samples that ought to have been classified as positive) divided by the number of accurate positive results. It measures how many genuinely positive cases are appropriately detected.

3.2 Setting of the lab

The system is implemented using the ANACONDA software, which is the most widely used data science platform and the cornerstone of contemporary machine learning ("Anaconda Distribution: https://www.anaconda.com/distribution/"). According to Anaconda, Inc. (2022), "We pioneered the use of Python for data science, champion its vibrant community, and continue to do so." Our story (retrieved from https://www.anaconda.com/our-story) is responsible for stewarding open-source initiatives that enable next developments. Using the Python programming language, the models are created. Numerous data analysis and visualization packages for classification and prediction are available in Python. A few of the packages are sci-kit-learn, pandas, NumPy, matplotlib, tensor flow, among others. The software is installed on a Windows computer running Python 3 of any version. Import the necessary Python packages, libraries, and dataset after Python has been installed. Following that, PyCharm's Integrated Development Environment (IDE) deploys and runs the program.

3.3 Materials

The dataset will be automatically created and will include more than 500 aberrant requests in addition to 1,000 typical queries. The HTTP requests are classified as being normal or abnormal, and the dataset contains threats like XSS, buffer overflow, and SQL injection.

3.4 Technical Context

This study aims to provide useful insights into the possibilities that abound in the field of Artificial Intelligence in creating automated responses to different forms of cyber threats. A major source of cyber threat is by way of malicious HTTP requests that can be sent to a HTTP server. To request a resource from a webserver, you send an http request via a browser, the http request is usually made up of a request line, set of header fields and a payload (or body) and method (e.g. . POST, GET). A webserver receives requests, validates these requests, and processes them, finally it sends a response back to the client(browser).

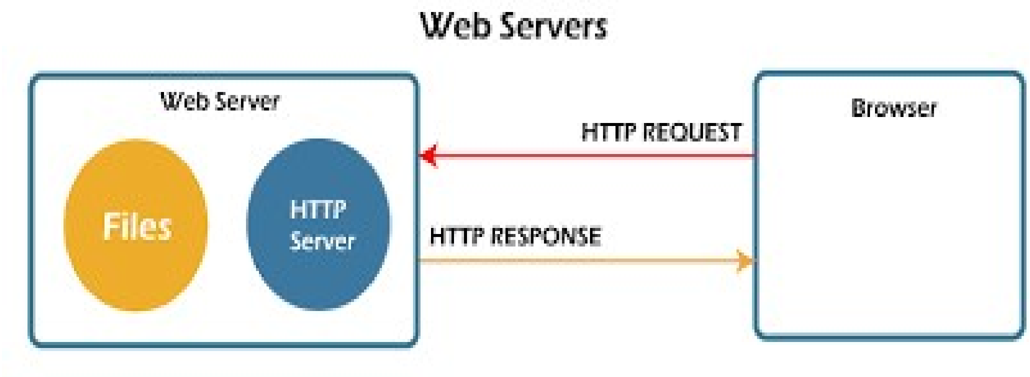

Figure 1 : web server request-response architecture
(Image by: https://www.javatpoint.com/web-servers)

More specifically we are interested in how machine learning algorithms can help to identify malicious http requests sent to server. For this study we have classified http requests into two categories namely: normal and anomalous http requests. A normal request is a http request that is formed based on the http standard and is free of any form of malicious interference or tendency. An example of a normal http request can be a user trying to get information from a specific web resource (GET /www.example.com/get_resource). A malicious request is a http request that is sent deliberately to attack, get information, or perform an action in an unauthorized manner. There are many forms of malicious http requests, but for this study we have focused mainly on three (3) forms namely: SQL injections, XSS attacks, and buffer overflow. The next section focuses on a more technical perspective as to how the machine learning algorithms are trained to detect malicious http requests.

3.5 Coding and Technical Implementations

The aim of this study is to predict how each of the four selected machine learning algorithms can detect malicious http requests. More specifically, we are interested in knowing how each algorithm can be able to accurately predict or categorize a normal request from a malicious request. To carry out this research activity efficiently and effectively, we performed certain prerequisites namely:

Data generation and ***wrangling***, ***training***, and ***testing*** of algorithms, and ***analysis of models' performance***. These prerequisites are itemized and explained in the sub-sections below.

3.6 Data Generation and Wrangling

We are interested in how well our machine learning algorithms can accurately predict http requests that are normal and which ones are anomalous, we decided to create a dataset that has two columns namely "***label***" and "***request***". The "label" column is the categorical column and contains the category of requests which is either "normal" or "anomalous". The "request" column contains the actual HTTP request payload and can also be normal or anomalous. We downloaded a robust array of the 3 selected forms of malicious http requests payload (SQL injection, XSS attack and buffer overflow). These payloads are downloaded from an open-source platform and are stored in CSV formats which afterwards we then automatically load these payloads iteratively (based on the number of requests specified in the code) in a random manner and stores it in a data frame.

3.7 Model Training and Testing

After generating the datasets, we proceeded to train each algorithm on the dataset. The training specifically aims to help each of these algorithms accurately classify normal http requests from anomalous http requests. To achieve this, we created four python classes that encapsulate the coding implementation for each of the four (4) machine learning algorithms. The subsequent sections will provide a detailed overview into the coding logic for the training and testing of each of the algorithms. Since we ensured that each of these algorithms follow the same sequence in the train_test_predict operation with some little differences in the types of parameters passed to each algorithm, hence we did not discuss the four (4) separately. However, we shall only discuss in detail the Random Forest Classifier.

3.8 Random Forest Classifier

To train the model using the random classifier algorithm, we created a class called ***RandomForestClassifierClass***. This class has a constructor that takes as a parameter the data frame we generated in the previous section. The class also has a method called train () that takes as a parameter an instance of the class. The train method or function encapsulates the actual training and testing logic, and returns a tuple of the four-performance metrics (f1, recall, precision, and accuracy) for further analysis and visualization.

The training and testing of the model are itemized in the steps below:

1. The data set is split into training and testing sets for both the labels(categorical) columns and the column we want to predict or categorize (the request column) using the train_test_split() function from sklearn's model selection library. This yields four (4) variables namely: X_train, x_test, Y_train and y_test
2. The training and testing categorical label columns (Y_train and y_test) are then encoded using the LabelEncoder class from sklearn preprocessing library. To do this we use the fit transform function of the LabelEncoder class to encode the training data and we use the transform () method to encode the testing data. Note that it is important for us to encode our data because most machine learning algorithms can only make train and predict numerical values (scalar or vectors).
3. The next step is to encode the column that holds the data we hope to predict. To achieve this however we must use a different and more complex type of encoder. This is because the LabelEncoder works best with ordinal data. i.e., data that takes on distinct values such as "normal" and "anomalous" using this kind of encoder will produce unexpected results since our request column can have textual data that can be quite unpredictable. To achieve this, we have used the TfidVectorizer feature extraction class which is the sklearn's implementation of the Term Frequency-Inverse document frequency measure.

Again, we call the fit_transform() and transform functions of the encoder to encode our training and testing sets respectively. Next, we applied some hyperparameters to fine tune the model's performance. This is done by creating an array of possible parameters that can be assigned to the Random Classifier algorithm, we then use the GridSearchCV to determine the best parameters that can be used with the classifier. The Random classifier class uses parameters such as n_estimators, max_depth, min_samples_split etc.

4. The GridSearchCV fit () function was used to fit the model. This function takes both the x_train and y_train as parameters.
5. Next, we use the predict () function to predict the labels of the data values on the basis of the trained model. This method takes the x_test as a parameter and returns the predicted labels y_pred.
6. We compared the y_pred from the previous step with the y_test to see how many of the values were predicted correctly. This is done using the metrics made available by sklearn: accuracy score, precision score, f1 score and Recall (there are other metrics, but we have streamlined these four (4) for this research. To further fine tune the performance of the model and test for underfitting or overfitting, we decided to use a cross validation technique. One such technique is the Leave -One-Out class from the sklearn library.
7. The Leave-One-Out cross validator provides train/test indices to split data in train/test sets. Each sample is used once as a test set set(singleton). It involves iteratively removing one data point from the dataset and using the remaining to train the model. The performance of the model is then evaluated based on the difference between the predicted values and the actual values for each left-out data point.

The steps (1-7) listed previously are iteratively performed and we store the results from each iteration. To comprehensively grasp the outcomes for each iteration, we created four arrays that appends the score for each metric at every iteration; namely: accuracy scores, precision scores, recall scores, f1_scores, we then find the average(mean) score of each metric and run our analysis on them.

3.9 Performance Evaluation

Utilizing the four (4) identified machine learning algorithms, the model was evaluated. This section demonstrates how the model behaved after being put into use. Each performance metric's meaning has been elucidated in detail in the preceding chapter. The precision result reveals what proportion of the things the classifier expected to be relevant actually are. The recall gives an indication of the number of items that the actually relevant classifier found. Here, the X_train and Y_train are fitted into the model before the prediction is made using the X_test. The accuracy_score, precision, recall, and f1_score is displayed to evaluate the outcome. These are used to assess how well our model is performing.

4. RESULTS

Every machine learning algorithm used in this investigation will be evaluated critically in this chapter, along with the outcomes. This evaluation involves taking a deep dive into the technical and mathematical intricacies of how each of the algorithms works. The section also aims to critically evaluate each of the metrics (f1 score, precision score, accuracy score and the recall score) that have been used to evaluate the performance of each algorithm. For this study, I have leveraged four (4) of the known machine learning algorithms; namely: Random Forest Classifier, Support Vector Machine (SVM), Decision Trees and KModes. Each of these algorithms have specific implementation in python, as provided by the sklearn machine learning library.

4.1 Model Performance and Analysis

To accurately measure the performance of the different models, we have leveraged four (4) of the known metrics used in machine learning and statistics namely: accuracy score, precision score, f1_score and recall score. The sklearn library provides the python-based implementations of these metrics. We will briefly describe again each metric and dive into the analysis of each metric for each of the four (4) algorithms.

Accuracy: It represents the proportion of accurate predictions to all input samples. In other words, it is the percentage of all correctly.

Precision: It is the ratio between the number of correctly positive findings and the number of positive results the classifier anticipated.
F1-Score: The accuracy of a test is evaluated using the F1-Score. The Harmonic Mean of memory and precision is the F1 Score. The F1 Score has a range of [0, 1]. It informs you of your classifier's precision (the proportion of instances it properly classifies) and robustness (the proportion of instances it does not miss).
Recall: It is calculated by dividing the total number of relevant samples (all samples that ought to have been classified as positive) by the number of accurate positive results. It measures the percentage of genuinely positive cases that are accurately identified.

Having looked again at the description of each metric, let us discuss the scores for the four (4) metrics for each of the algorithms.

The data sets used for the training and testing of our four algorithms models were done using two different sizes (i.e., size one and size two).
Size One: 10 randomly generated anomalous and 20 randomly generated normal payloads.
Size Two: 100 randomly generated anomalous and 200 randomly generated normal payloads.

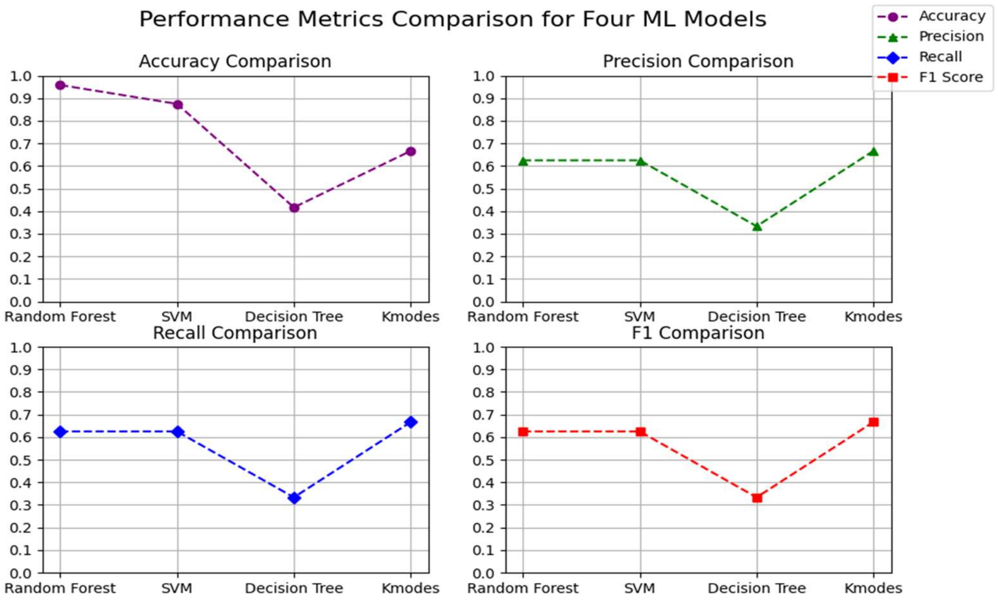

Figure 2: The line graph above shows each metric score for the four (4) machine learning algorithms for the first size (10 anomalous and 20 normal HTTP requests)

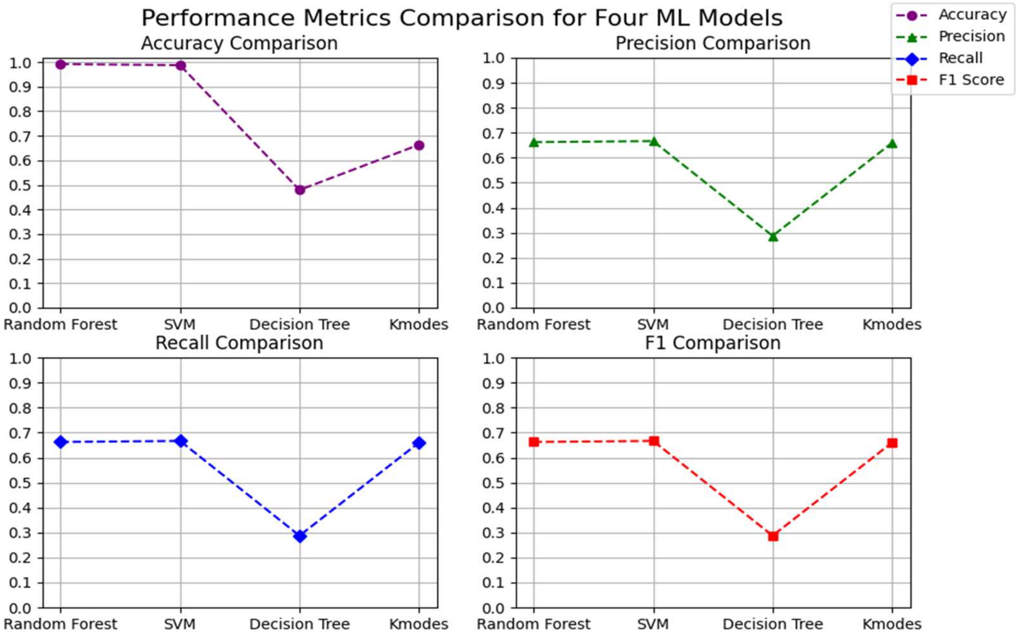

Figure 3: The line graph above shows each metric score for the four (4) machine learning algorithms for the second size (100 anomalous and 200 normal HTTP requests)

4.2 Summary of the Models Performance

The preceding sections show us two diagrams that depict briefly the performances of the four (4) selected algorithms for the two data sizes using the four (4) metrics.

Accuracy Score: For the first data size, it is observed that the Decision Tree classifier has the highest accuracy score of 0.9, followed by the Support Vector Machine with an accuracy score of about 0.85, the Random Forest Classifier has an accuracy score of about 0.83 with the Kmodes having the lowest accuracy score of 0.6.

Precision Score: For the first data size, the Support Vector Machine, Decision Tree Classifier, and KModes all have a precision score of roughly 0.67 while the Random Forest Classifier has a lower precision score of 0.65

Recall Score: For the first data size, the Support Vector Machine, Decision Tree Classifier, and KModes all have a recall score of roughly 0.67 while the Random Forest Classifier has a lower precision score of 0.65.

F1 Score: For the first data size, the Support Vector Machine, Decision Tree Classifier, and KModes all have an f1 score of roughly 0.67 while the Random Forest Classifier has a lower precision score of 0.65.

For the second data size, we have an accuracy score of for the Random Forest Classifier, SVM, and Decision Tree Classifier 0.98 for each of them and 0.667 for the KModes, a precision, recall, and f1 score of around 0.667 for all the four (4) algorithms.

5. CONCLUSION AND DISCUSSION

The aims and objective of this study is to correctly predict how each of the four selected machine-learning algorithms can detect ***malicious HTTP requests***. This research emphasizes how AI methods like machine learning can be used to enhance incident detection skills. To identify potential security incidents, researchers have investigated the use of AI algorithms to monitor network traffic, system logs, and security event data correctly and instantly. More specifically, we are interested in how each algorithm can be able to accurately predict or categorize a normal request from a malicious request. The results presented by this study have shown how Artificial Intelligence and Machine learning can enhance the capacity of cyber security experts in combating cyber-attacks, specifically attacks in the form of malicious HTTP requests. In this study, we selected four (4) machine learning algorithms to help us train models that can classify or predict malicious HTTP requests. The algorithms selected are: The ***Random Forest Classifier***, ***Support Vector Machine***, ***Decision Tree Classifier*** and ***K-Modes***. To have a more comprehensive test and analysis, we tested the four (4) algorithms with two data sizes and used different hyperparameters to fine-tune each of the models. Further fine tuning was done using a ***k-fold cross validation*** technique. For the first test size the ***Decision Tree Classifier*** appears to be better in classifying ***anomalous http requests*** with an accuracy score of about 98%, while the other metrics (precision, recall and f1 scores) seem to be constant at 67%. For the second test size, which had more data points, the accuracy scores of the ***Random Forest Classifier*** and the ***Support Vector Machine*** took an upward dive to get a 98% score together with the Decision Tree Classifier (which did not seem to increase in performance with the increase in training data). It is worthy of note that the KModes classifier had an average of about 65% accuracy across all four metrics (accuracy, precision, recall, and f1 scores).

5.1 Limitations

Even though this study provides us with useful insights as regards to how different machine learning algorithms can help to detect cyber threat, it is however not free of some limitations. Machine learning requires large amount of data to create efficient models, training these models on these data sets can be quite resource intensive, furthermore improving the performance of these models requires some extra finetuning like the GridSearchCV and Leave Out One implementation which proves to require high performance computing. In addition, because most of the datasets used are historical in nature, there is tendency that these models might not be able to make accurate predictions on current data sets.

5.2 Recommendations for Further Research

This study has exposed us to the possibilities that abound in the field of Data Science in combating cyber-threats. More Specifically we have been able to see how we can train learning and predicting models using four (4) different machine learning algorithms to detect and categorize malicious HTTP requests.

However, this is a relatively new focus in the field of Data Science and there are still more grounds to cover. Below are recommendations for improvements for this study or further research.

- Further research should be done on how other Machine Learning Models not covered in this study can be useful in combating cyber threats.
- Further research should be done on how these Machine Learning Models can be used to classify other forms of cyber threats like DNS tunnelling, Zero-day exploits, Denial of Service attacks and Man in the Middle attacks.
- Further research can be done on designing a machine learning based approach for identifying and mitigating insider threats in an organization.
- It is also important to further research on ethical considerations in the use of Machine Learning for Cyber-attack mitigation.

Using the information outlined in the summary above, we can conclude that the Decision Tree Classifier appears to be the most efficient algorithm for predicting malicious HTTP requests, however, there seems to be a significant difference between the accuracy score and the other metrics used in evaluating the performance of the model. We have also observed that increasing the data size has a significant impact on the accuracy score of the Support Vector Machine and the Random Forest Classifier, we saw the percentage rise to about 98%, however, this increase in data size did not have any significant impact on the scores of the other metrics. Also worthy of note is the fact that the K-Modes classifier seems not to be affected by the data size with all four (4) metrics remaining constant at a 65% accuracy. Generally, we can infer from the analysis provided by this research that Machine Learning can be instrumental to developing more effective and efficient counter-attack measures in combating cyber threats.

REFERENCES

Abou-Assaleh, Tony, et al. "N-gram-based detection of new malicious code." *Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004.* Vol. 2. IEEE, 2004.

Barker, Charity. "Applications of Machine Learning to Threat Intelligence, Intrusion Detection and Malware." (2020).

Obunadike, C., Adefabi, A., Olisah, S., Abimbola, D. and Oloyede, K. (2023) Application of Regularized Logistic Regression and Artificial Neural Network model for Ozone Classification across El Paso County, Texas, United States. *Journal of Data Analysis and Information Processing*, **11**, 217-239. doi: 10.4236/jdaip.2023.113012.

Berninger, M., and A. Sopan. "Reverse Engineering the Analyst: Building Machine Learning Models for the SOC." *Internet: https://www. FireEye. com/blog/threat research/2018/06/build machine-learning-models-for-the-soc. HTML* (2018).

Elovici, Yuval, et al. "Applying machine learning techniques for detection of malicious code in network traffic." *KI 2007: Advances in Artificial Intelligence: 30th Annual German Conference on AI, KI 2007, Osnabrück, Germany, September 10-13, 2007. Proceedings 30*. Springer Berlin Heidelberg, 2007.

Fan, Yujie, Yanfang Ye, and Lifei Chen. "Malicious sequential pattern mining for automatic malware detection." *Expert Systems with Applications* 52 (2016): 16-25.

Hou, Yung-Tsung, et al. "Malicious web content detection by machine learning." *expert systems with applications* 37.1 (2010): 55-60.

Islam, Chadni, Muhammad Ali Babar, and Surya Nepal. "A multi-vocal review of security orchestration." *ACM Computing Surveys (CSUR)* 52.2 (2019): 1-45.

Lazic, Ljubomir. "Benefits from AI in Cyber Security." *The 11th international Conference on Business Information Security. Belgrade, Serbia*. 2019.

Mohasseb, Alaa, et al. "Predicting CyberSecurity Incidents using Machine Learning Algorithms: A Case Study of Korean SMEs." *ICISSP*. 2019.

Morgan, Steve. "Cybersecurity talent crunch to create 3.5 million unfilled jobs globally by 2021." *Cybercrime Magazine* 24 (2019).

Olufemi, I., Obunadike, C., Adefabi, A. and Abimbola, D. (2023) Application of Logistic Regression Model in Prediction of Early Diabetes across United States. International Journal of Scientific and Management Research, 6, 34-48. doi:10.37502/IJSMR.2023.6502

Sarker, Iqbal H., Md Hasan Furhad, and Raza Nowrozy. "Ai-driven cybersecurity: an overview, security intelligence modeling and research directions." *SN Computer Science* 2 (2021): 1-18.

Shabtai, Asaf, et al. "Detecting unknown malicious code by applying classification techniques on opcode patterns." *Security Informatics* 1.1 (2012): 1-22.

Shaukat, Kamran, et al. "A survey on machine learning techniques for cyber security in the last decade." *IEEE Access* 8 (2020): 222310-222354.

Trifonov, Roumen, Ognyan Nakov, and Valeri Mladenov. "Artificial intelligence in cyber threats intelligence." *2018 international conference on intelligent and innovative computing applications (ICONIC)*. IEEE, 2018.

Tyugu, Enn. "Artificial intelligence in cyber defense." *2011 3rd International conference on cyber conflict*. IEEE, 2011.

Wang, Tzu-Yen, et al. "A surveillance spyware detection system based on data mining methods." *2006 IEEE International Conference on Evolutionary Computation*. IEEE, 2006.

Zhang, Boyun, et al. "Malicious codes detection based on ensemble learning." *Autonomic and Trusted Computing: 4th International Conference, ATC 2007, Hong Kong, China, July 11-13, 2007. Proceedings 4*. Springer Berlin Heidelberg, 2007.

Arpitha, B., et al. "Cyber Attack Detection and notifying system using ML Techniques." *International Journal of Engineering Science and Computing (IJESC)* (2021).

MCITP, MCSE. "The risk of artificial intelligence in cyber security and the role of humans." *Texila International Journal of Academic Research*. 2520-3088. (2019).

Apruzzese, Giovanni, et al. "The role of machine learning in cybersecurity." *Digital Threats: Research and Practice* (2022).

Goni, Ibrahim, et al. "Cybersecurity and cyber forensics: machine learning approach." *Machine Learning Research* 5.4 (2020): 46-50.