National College of Ireland

A Game Theoretic and Machine Learning Approach for Strengthening Network Security Using Honeypots in Healthcare

MSc Research Project

MSc Cybersecurity

Iqra Fareed

Student ID: x23163461

School of Computing

National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ..... **Iqra Fareed**.....

Student ID: ..... **X23163461**.....

Programme: ..... **MSc Cybersecurity** ..... **Year:** .....**2024**.....

Module: .....**Practicum**.....

Supervisor: .....**Vikas Sahni**.....

Submission Due Date: .....**16/09/2024**.....

Project Title: **A Game Theoretic and Machine Learning Approach for Strengthening Network Security Using Honeypots in Healthcare**

Word Count: .....**8770**..... **Page Count**.....**20-22 pages**.....

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: *Iqrafareed*

Date: **16/09/2024**

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on a computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Game Theoretic and Machine Learning Approach for Strengthening Network Security Using Honeypots in Healthcare

Iqra Fareed
x23163461

Abstract

Detrimental effects of ransomware attacks on healthcare sectors continue to grow in prevalence, threatening patients, data privacy and essential operations in healthcare organisations. This research focused on establishing the optimum defence strategy for protecting healthcare networks against ransomware attacks with the help of machine learning, honeypots, and game theory. Using Gradient Boosting Machines (GBMs), the machine learning model is employed to predict the presence of ransomware by learning and analysing features in the network traffic. Moreover, the framework of Bayesian game theory was employed to improve and evolve network defence dynamically depending on probabilistic threats. These complex technologies have aimed to provide an intelligent and dynamic security environment to counter ransomware threats efficiently besides detecting them. It obtained a high detection rate of ransomware, demonstrated the utility of honeypot in acquiring information on attack behaviours, and the real-time reinforcement of defence mechanisms.

1 Introduction

Cyber threats have been emerging rapidly in the digital world, with ransomware attacks being one of the significant cybersecurity concerns in the current world. Ransomware is malicious software that locks the victim's files and denies them access unless a specific amount of money is paid. Another factor that makes ransomware attacks more dangerous is that even after paying the ransom, one cannot be sure that the data will be decrypted, adding to the losses incurred. This issue is most acute in healthcare industries, which involve managing patient information and providing critical services. Today's threats are more frequent and complex, requiring robust and flexible tools to safeguard core assets and data.

Ransomware attacks have been focusing on HCOs (Healthcare Organizations) because such organisations store valuable data and work to offer essential services. In October 2022, Australian healthcare institution Medibank and its employees were held hostage by a group of Russian hackers linked to the REvil ransomware gang[2]. The attackers swiped the personal details of nine million users of a social networking site in Southeast Asia—7 million people, including political leaders like Australian Prime Minister Anthony Albanese and Minister Clare O'Neil. Paczar stole information consisting of name, passport number, date of birth, social security number, and medical records. This case of Medibank clearly showed that ransomware attacks had the dangers of financial loss and reputational damage, as exemplified by the refused 10-million-dollar ransom demand from Medibank. Also, in May 2021, the Ireland Health Service Executive (HSE) was hit by ransomware, which paralysed most of its healthcare operations in the country[1]. Traditional security tools are generally not very helpful in detecting and subsequently addressing these threats and exploits, as most are non-proactive. Therefore, the focus is shifted to developing new security measures that can

[1] https://www2.hse.ie/services/cyber-attack/what-happened/

[2] https://www.smh.com.au/politics/federal/identity-of-medibank-hacker-confirmed-government-invokes-cyber-sanctions-20240123-p5ezbl.html

effectively address potential risks that might threaten the system even before the risk manifests itself. This research aims to address this gap by presenting an ideal architecture that consists of learning algorithms, honeypots, and game theory.

This study aims to improve the security of healthcare networks from ransomware attacks. This is accomplished through the following: the training of a machine learning model with Gradient Boosting Machines (GBMs) to quickly recognise ransomware attacks from network traffic, the incorporation of honeypots into the overall healthcare network to lure and analyse ransomware attacks, as well as the use of Bayesian game theory to continuously fortify the defences of the network as per probabilistic models of the threat. Furthermore, the location of honeypots on the network is intended to provide maximum visibility of the activity and minimal interference with the attack.

For this research, it is essential to state that integrating the machine learning system with honeypots in the healthcare facility could help detect early ransomware attacks and prevent them before compromising the protected medical data. Employing Bayesian game theory will enable the security system to evolve to counter what is new and, therefore, will help the system become more secure. Consequently, this study outlines clear recommendations and concrete models, which can be applied directly to specific healthcare networks — an approach that addresses an existing critical problem in a scalable and efficient manner.

In applied mathematics, game theory is a set of rules that describes the competition between two or more participants depending on the other participant's actions. It is helpful in cybersecurity since it contributes to elaborating the necessary strategies to understand and predict the behaviour of network attackers and protectors. Cybersecurity can classify these interactions as cooperative or non-cooperative, zero-sum or non-zero-sum games. With cooperative games, the players interact on one side, while with non-cooperative games, the players may or may not interact. The concepts that are used in the game theory include the players who are the decision-makers, strategies that are the course of action that is chosen by the players, payoff, which is the gain or loss that results from the strategies that have been selected as well as equilibriums which is the situation that cannot change even if one of the players changes their strategy without consulting the other players.

The research used a well-structured approach that included data gathering and analysis, using ransomware datasets to train and test machine learning models, building and enhancing GBM models for ransomware identification, effectively placing honeypots in the healthcare network for data accumulation and analysis of the attacker's pattern, and using statistical models to improve the protective mechanisms perpetually.

The rest of this paper is organised so that the first part covers the research background and problem statement, the objectives set, the importance of the study, and the methodological approach adopted in the research. Then, critically analysed peer-reviewed literature on honeypots, machine learning, and game theory applied to security in the literature review section. The 'Methodology' section in the present proposal outlined the plan for devising and implementing the security framework. This is followed by describing how the research has been practically applied by describing the implementation and testing of models leading to honeypots. The results and discussions focus on evaluating the solution's effectiveness and its comparison with other approaches. Last, the conclusion and future work section briefly discusses the results obtained, their usefulness, and a proposal on what can be done.

[1] https://www2.hse.ie/services/cyber-attack/what-happened/
[2] https://www.smh.com.au/politics/federal/identity-of-medibank-hacker-confirmed-government-invokes-cyber-sanctions-20240123-p5ezbl.html

2 Related Work

2.1 Honeypots

These are systems that draw in the attackers to analyse them. A paper by (Selvaraj et al., 2016) expounds on integrating honeypots with Intrusion Detection Systems to improve DDoS attack detection. They included their work, a network with an anomaly detection system with very high detection rates. In addition, there are a few issues about the study; one of the most noticeable is that the results may be hard to use in real life. In their research, (Yeldi et al., 2016) deployed a Mirage honeypot system in an education setting. This research emphasises the effectiveness of the honeypot in conveniently tracking diverse modes of attack. This is, however, only a pilot study, and the testing period is slightly short. There is a need for further studies that will enhance understanding of the system's sturdiness. (Thakar, 2005) provides details of HoneyAnalyzer-a, a tool specialising in analysing and extracting Intrusion Detection patterns from honeypot logs. It also makes it easier for the security administrators to use the solution, primarily through the graphical interface, due to its ease of user-friendly interface. It still frees the solution from large-scale automated usage since much depends on manual intervention.

Most new developments for honeynets must address architectural advances to increase effectiveness and deployment options successfully. (Krishnaveni, Prabakaran, & Sivamohan, 2018) Give a detailed overview of honeynets considering their development and usage in cloud systems. The information follows: They support using honeynets and resource containers such as Docker, stating that the two should be deployed hand in hand. However, their work needs to provide strong empirical support that these results can be applied in many other areas of operation. Low, medium and high interaction honeypots are treated with thorough comparative analysis provided by (Han et al., 2016), where the authors consider the type of honeypots and their relative strengths (Jiang et al., 2011), describing possibilities of using high and low interaction honeypot systems and attempts to use a mid-range honeypot to provide the advantages of each while reducing the drawbacks. This approach coincides with (Nawrocki et al., 2016), who stress the presence of dynamic honeypot systems, which will change the conditions of its work depending on the threats encountered during its operation.

This paper explains that reputation-based systems employ honeypots to enhance network management and administration and develop detailed profiles of attackers' characteristic tendencies. Another study by (Patel et al., 2020) proposes using honeypots to manage and detect persistent threats, which, in turn, allows for the modification of defences. This method improves the provision of blocking the mischievous participants before they cause havoc because it depicts how they have been behaving in the past.

2.2 Machine Learning & Honeypots

Machine learning has proven to be sociable in cybersecurity but is mainly used for ransomware detection. (Khurana, 2023) describes the use of Algorithms and Machine learning to detect ransomware threats with learning-based concepts that help make the model fluent and capable of learning from past experiences. A recent study by (Ahmad et al., 2023) highlights the status of machine learning in ransomware detection and insists on the efficacy of feature extraction and model optimisation. (Alraizza & Algarni, 2023) Examined the history of ransomware detection through deep learning approaches and thus viewed the world's improvement in detecting intricate ransomware patterns. More recently, (Selvaraj et

al., 2019) enhanced the methodology by integrating Holt-Winters forecasting, Genetic Algorithms, and fuzzy logic to serve secured software-defined networks (SDNs) against DoS/DDoS attacks. This underscores the growing role of machine learning in cybersecurity and the need for further research in this area. Similarly, (Patel et al., 2020) propose a novel approach for employing honeypots in managing reputation by identifying stable threats and modifying the protection mechanisms accordingly. It helps improve the probability of preventing the lousy actor based on its behaviour pattern that has been identified in the past. This highlights the ongoing development in cybersecurity and the need for further research, particularly in the context of honeypots and machine learning.

2.3 Game Theory & Honeypots

This approach has laid the foundation for practical cybersecurity measures, especially using game theory. A study by (Krishnaveni et al., 2018) overviews honeypot systems based on game theory, outlining their suitability in identifying and responding to advanced attacks. Although they have provided a detailed study, their work needs to explore computational complexity and scalability problems effectively. (Roychowdhary et al., 2020) devise a learning-based game for UAV networks that would enable collaborative intrusion detection. This strategy is original, but the possible delay and communication overhead involved in constant information sharing must be addressed. (Kyung et al., 2017) further, expand on considering honeypot defences in SDN environments and point out that the collaboration style is vital to improving security.

Moreover, (Zhu et al., 2021) investigate defensive deception strategies using game theory and machine learning. They comprehensively analyse how these two sophisticated approaches can be used to identify and prevent cyber threats in the scientific literature. The survey also includes a range of game theoretic models and machine learning used to design and develop reactive and preventive defence techniques. These strategies are described by Zhu et al. as possibly improving the modernity and efficiency of such systems and the methods of protection where options can change dynamically depending on the attacker's actions.

Strengthening other security layers through the honeypot approach and game theory-based models is critical for the advances of contemporary cybersecurity. When implemented with robust IDPS tools, the mixture of these methods establishes multiple barriers against these threats. This integration aids in knowing the aggressor’s conduct, devising the most suitable defence approaches, and preventing threats. The more extensive defence mechanisms, like the mentioned objectives, help improve the viability of network security systems.

Cyber-attacks, especially DDoS attacks, can be managed using a strategic approach provided by game theory. Bedi et al. are specific with game-theoretic models, which encompass defenders who seek to put up their firewall to the best they can against the attackers who seek to exhaust bandwidth. This model uses the Nash equilibrium to determine which defensive strategies would be most preferable; this model was relatively successful in numerous simulations (Bedi et al., 2011). Likewise, Wang et al. play strategic game theorists in a smart grid by using honeypots to eliminate DDoS. This way, melee is minimised in terms of deployment costs while simultaneously maximising the benefits of gaining intelligence on the attackers, managing resources, and enhancing the existing defence arrangements. Zhu and his colleagues expand on defensive deception strategies from a game theory and machine learning perspective, thus stressing the approach's viability when dealing with sophisticated cyber threats (Wang et al., 2017).

Appropriate IDPS tools preserve the network's structural and informational integrity. Unlike many other papers that compare and review IDPS tools, Wahyu et al. 's work concentrates on the inspecting and preventing functions of these tools. Practical IDPS tools must be implemented to enhance the general security required to strengthen the methods that detect the three peculiarities of security threats. Further, Bedi et al. proposed a study for the defence mechanisms based on game theory against DDoS attacks on TCP/TCP-friendly flows, which can contribute to the solution with how IDPS can be integrated into the game theoretic models to improve the defence against the bandwidth depletion attacks (Bedi et al., 2011).

2.4 Interaction Based Honeypots

Honeypots can, therefore, be regarded as primarily fake systems which aim at enticing the attacker; after reviewing Naeem's work that gives a detailed analysis of honeypots and classifies them based on utilisation for research or production purposes and interaction level: low, medium, high. Researching honeypots aims to provide information about specific malicious actions and help design effective security systems. However, they are pretty complicated and tend to be expensive to maintain. On the other hand, production honeypots are more accessible to develop and can be protective immediately, though they give less information about the attacker's plans (Naeem, 2021). (Wang et al., 2017) Further, this approximation is achieved by surveying honeypots and honeynets dedicated to the IoT, Industrial IoT, and Cyber-Physical Systems, indicating their relevance to those novel areas.

This section has noted the recent development of solutions for strengthening network security. (Prabowo et al., 2023) It stresses the importance of IDSs and IDPs for protecting the network infrastructure. Such an assessment emphasises the effectiveness of different IDPS tools and the best way to identify security threats promptly. Thus, the research emphasises the role of incorporating such tools into the network architecture to form a layered structure that should help combat different cyber threats.

In addition, this study is complemented by the work collected by (Franco et al., 2021), which presents a recent systematic review of honeypots and honeynets within IoT, Industrial IoT, and cyber-physical systems. Their paper also focuses on the use and efficiency of these security tactics in enticing and observing malicious acts. The authors discuss different honeypot categorisations—low, medium, and high interaction honeypots—strengths and weaknesses. They endorse using honeynets in conjunction with other security solutions to maximise the identification and prevention of threats, especially considering the IoT and industrial systems that are not easily protected using conventional security methods.

3 Research Methodology

This research demonstrated that the emerging healthcare network's defence layout comprised complex machine learning, honeypot deployment, and game theory interconnections. This narrative guided the development of these components into a coherent strategy against ransomware attacks.

3.1 Machine Learning Model Development

The first step in training a machine learning technique that can help detect ransomware attacks is data acquisition. This includes downloading different Android applications, including regular and those with ransomware, to which fresh samples from cybersecurity organisations are linked. A feature of this diverse ground set is that the model will be duly prepared to work on a range of ransomware attack types.

It is analogous to the basic preparation of the raw materials that the data collected for analysis would undergo. This comprises noise removal, record elimination, record imputation, feature scaling, and feature transformation; specific techniques include one-hot encoding. The gathered data is divided into training, validation, and test sets for proper model development.

Thus, the next step involves Exploratory Data Analysis (EDA), which is used to identify the nature of the dataset and data dispersion, its mean, median, mode range, etc. Histograms, box plots, and scatter plots serve as means of data characteristics and relationships to advance the analysis.

In the same way, after EDA is done, feature engineering is applied, such as carving and polishing the statue. In this step, one chooses and precisely determines only the most essential characteristics and utilises methods like Principal Component Analysis (PCA) to reduce the number of features and preserve as much information as possible. Secondary computed variables, like the average of the total network traffic, are used to increase the quality of information and avoid passing unnecessary data to the model.

The last stage in the model is identifying and tuning the machine learning model. Five classifiers, namely gradient boosting, random forest, logistic regression, k-nearest neighbours, and decision trees, are measured for their accuracy, precision, recall, and F1 score. The classifier that was classified as the best is a Gradient Boosting Classifier because it proved its efficiency by being accurate and adaptable to data containing many variables.

3.2 Honeypot Deployment in Conceptual Network Design

When such a model is available to the researcher, the next chapter entails positioning the honeypots tactically in a theoretical healthcare network based on ISO/NIST (International Organization & standardisation/ National Institute of Standards & Technology) standards. These honeypots are like decoy ducks, used to draw ransomware attacks to them instead of the actual operating systems. The network is compatible with the NIST and ISO policies necessary for sound security systems and proper honeypot location.

In the external environment, there are open services that, together with honeypots, are placed in the Demilitarized Zone (DMZ) and are designed to lure external attackers. Within the network, honeypots are placed close to critical assets such as Electronic Health Records (EHR) systems to identify malicious insiders. IoT and medical device networks have adopted a honeypot in which the potential attackers are given imitations of vulnerable devices. For this reason, the honeypots are well-positioned to alert the administrator of unauthorised activities without necessarily interning with legitimate network operations.

4 Design specifications.

4.1 Network

The conceptual Healthcare Network Architecture was based on compliance with NIST SP 800-53 Rev. 5 and ISO/IEC 27001:2013 (NIST Special Publication), frameworks to build a multi-layered security and monitoring system for safeguarding healthcare information and maintaining business continuity. The router formed the backbone of the design, through which all the transmitted and received traffic passed. Routing was critical for managing where the data traffic should go to the right segments of the network to reach the target destination. On the positive side, it provided control of traffic flow from a single central point and the option to apply global security policies. However, its downside was that it could become a massive single point of failure if not correctly designed with redundancy.
The segmentation was done through standard and additional firewalls, both internal and external, to control and monitor the traffic within the various segments. These firewalls offered basic protection against such threats, acted as a strong barrier against unauthorised access, boosted networking segregation and offered strict control of data traffic, thereby boosting its security from internal and outside threats. However, they can turn into a node of concentration if poorly designed, and they can cause difficulties with setting firewall rules.

VPN gateways were used to connect users securely to the network from the internet. This enhanced privacy and ensured that an operation would be performed on the remote device bypassing the local device; however, this comes at the cost of a performance overhead incurred through encrypting and decrypting messages. In particular, the authentication of a VPN was an issue because the exposure of users' credentials threatened security.

Honeypots were deployed in a DMZ network, the internal LAN of an organisation, and the IoT Internet network. Most of them were employed to gain a better insight into attacks and bolster a network's general security. Honeypots helped in the early detection of threats and offered a detailed overview of the activities carried out by attackers, hence enhancing threat intelligence. However, if they fail to isolate, honeypots could become a starting point for real attacks on systems, and as a result, additional time and effort are needed in their management.

Specific segments like Identity and access management (IAM) systems, Intrusion detection systems (IDS), network monitoring tools and backup and recovery segments were still core to the network structure. IAM systems served as the means for managing user identities and their rights concerning the data. IDS consistently monitored the network for incidences of security compromise, while network monitoring programs yielded information on the state of health of the network. Business continuity systems of all firms allowed for rapid recovery of lost or compromised data. These elements extended security features through continuous monitoring, limited user access, and fast data recovery, but at the same time, they fuelled network densification.

Within the access layer, clients such as mobiles, IoT intelligent medical devices, and workstations were connected to the internal network. These devices communicated with application servers and databases, another system called the EHR system, and others based in the data centre. This configuration helped guarantee that patient data was readily available and easily navigable by healthcare providers. This integration enhanced organisational functionality, reduced costs, and availed information, but at the same time, it needed high defence mechanisms against any malicious attacks.

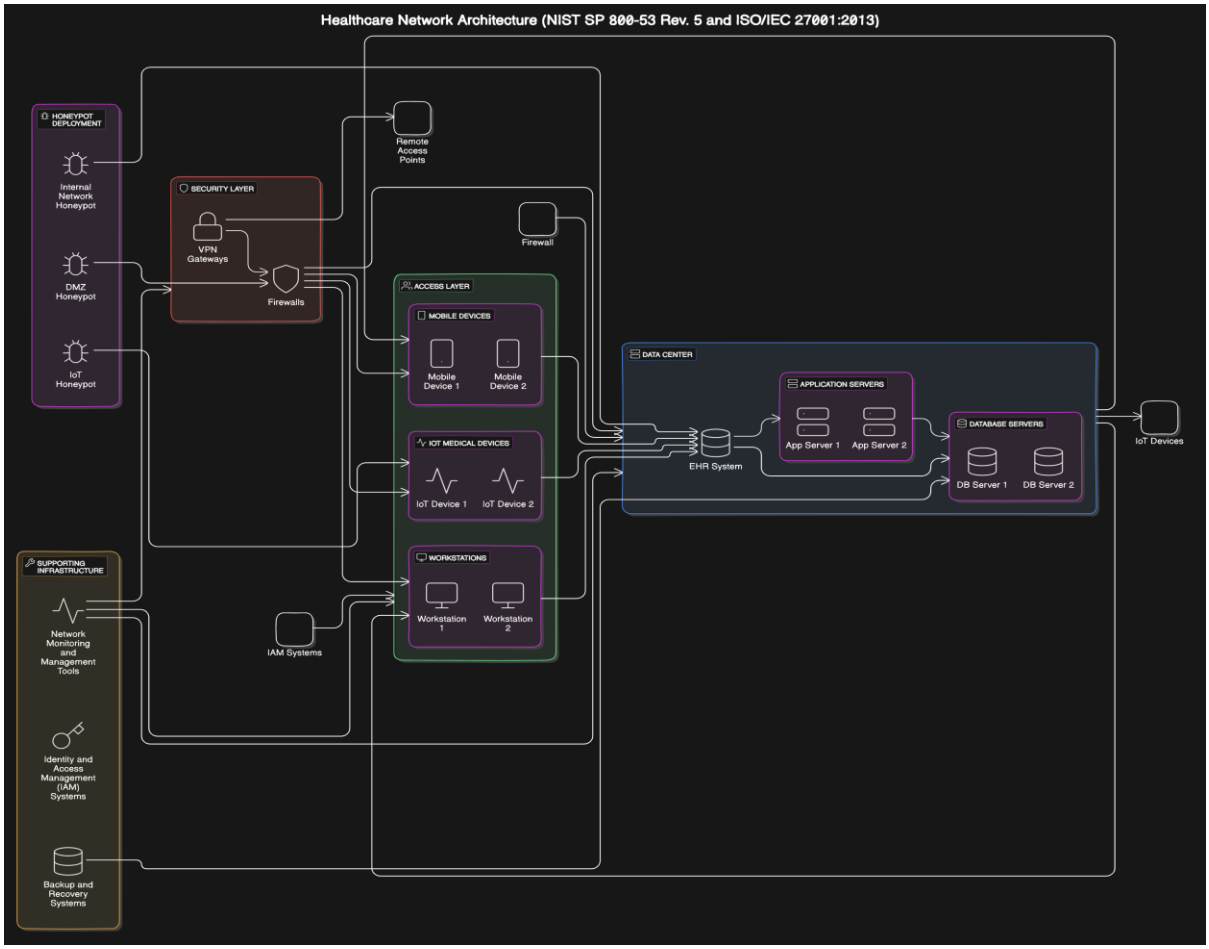

Figure 1: Honeypot Deployment

4.2 Techniques & Frameworks

The infrastructure of cybersecurity defence was offered as an approach based on machine learning and game theory. This framework was about proposing a cloud-based solution that would be able to detect, study and avoid ransomware attacks on healthcare networks. The model served as the foundation for the Random Bytecode Compiler's (RBC) approach as it leveraged the GBMs technique owing to its efficiency in analysing multivariate data identified within the organisation. Samples were collected from 1129 Android applications, including ransomware and ordinary applications, and completed with additional data from cybersecurity companies. The collected data were pre-processed by removing noisy data, normalising the range of features, and transforming categorical variables to numerical variables using techniques such as one hot encoder. The pre-processed data was split into training, validation, and test sets for the correct model building.

Apart from the Machine Learning model, honeypots were placed at suitable positions in the healthcare network to trap and study ransomware attacks. These honeypots mimic the weakest systems and services and tempt the attacker and, in the process, log every detail of the behaviour. When honeypots are allowed around EHR and other critical structures in the DMZ, one can identify external and internal threats as they conduct their operations without any hindrance. This setup is relatively active, which is beneficial when comprehending the attackers' actions and enhancing protective means afterwards.

The Bayesian game model has been created to support enhancing the defence strategy. In this model, the defender can engage in machine learning-based or standard defence while the attacker who chooses to attack is the other player. Pay-offs for the players and the strategies followed are determined based on the probability of attack. The designed model aims to get the Nash equilibrium; this is where the two players have no benefit by altering their strategies on their own. This equilibrium updates the probability of an attack online using the Bayesian technique to identify the right defensive strategy.

4.3 Outputs produced

The project yielded several main products. The first step is to standardise the data's features and format it for Machine learning algorithms. Then, data preprocessing is done. This cleaned and structured data is crucial in preparing the model for its training process.
Then, the Gradient Boosting Machine model was built and fine-tuned to detect ransomware. Other factors assessed included the accuracy, precision, recall, and F1 score, demonstrating the model's capacity to detect ransomware threats effectively.
Also, code was generated, with Python being the most frequently used language in completing the project's different parts. This comprises data preparation, model building, and model assessment. For data handling techniques, Pandas and NumPy were used; for machine learning algorithms, Scikit-learn was used; and for data visualisation, Matplotlib was used.
Collectively, this output supports a firm and adaptive defence system that can manage ransomware threats in healthcare networks.

5 Implementation

5.1 Tools and languages

Google Colab was used as the Python environment for this project. The primary language of this project is Python, as this language is somewhat universal and has rich support in terms of libraries. A set of several vital libraries was employed to achieve the stated objectives. Pandas and NumPy were highly helpful in the data manipulation and analysis. Both would have beneficial tools for handling large databases. The scikit-learn package was also used to construct and train the machine-learning model because it provides a vast list of algorithms and options for assessing the best match. Matplotlib was used to perform data visualisation; this tool allows the creation of easily readable visualised data suitable for supporting exploratory data analysis and enhanced data presentation.

5.2 Model Training

5.2.1 Objective

The primary objective of the model training phase was to develop a machine learning model with a high accuracy rate in identifying ransomware. This model was designed to discern several attributes of ransomware, distinguishing between normal and malicious behaviour. The ultimate goal was to maximise the detection level while minimising false alarms, ensuring the model's reliability and near-perfection in real-world applications.

5.2.2 Data collection

Data collection is an essential step that must be carried out during model training. It entails a collection of Android applications set – a mix comprising normal and ransomware applications[3]. Besides, the sources of real-labelled datasets for machine learning research in

[3] https://www.kaggle.com/datasets/subhajournal/android-ransomware-detection

databases are also included. In addition to the set of samples described above, more samples collected from groups of cybersecurity researchers and projects in the open domain are incorporated into the dataset. Taking large samples from various organisations promotes generalisation and improves the model's capability to identify ransomware.

5.2.3 Data Preprocessing

Concerning data cleaning, it was also a part of data preprocessing, where mainly unwanted noise signals were usually eliminated. The procedures for this process are mentioned below. The data in the dataset was preprocessed through data cleansing; duplicates were removed, while records with missing values were handled appropriately before modelling. Next, adjusting the features to a standard degree, for example, is normalised between (-1, 1) or (0 and 1), so converging as the model is trained takes little time. Categorical data collected were also converted to a numerical format that was goodly received by the various learning algorithms, such as through one-hot encoding. Lastly, the assembled dataset was then separated into a training set and a testing set, in most cases, with a 70/30 ratio to specifically deal with the issue of testing the model on the unseen data. It was essential to achieve it to evaluate the model's generalisation capacity.

5.2.4 Exploratory Data Analysis (EDA) & Feature Engineering

Exploratory data analysis (EDA) methods were used to determine the dataset's characteristics and decide subsequent model-building actions. The EDA method touched on basic mathematical computations with the dataset measures, including the mean, median, and standard deviation. For distributions, outliers and relationships in the features, histograms, box and whisker plots and scatter and pair plots were used. The association of class and features was established to look for the possible feature overlap or similarity with another kind of feature, which led to transformations when necessary. By performing EDA, the distribution of data and any issues of concern that need to be resolved before proceeding with the model training came to light. The feature engineering process creates new features from the existing datasets or transforms existing features into new forms for better model prediction. This process provided reference points for defining the characteristics of variables to be used in the models' input data.

5.2.5 Model Selection & Training

Model selection can also be called algorithm selection in practical terms and implies determining which learning algorithms are appropriate for the relative characteristics of the data set under consideration. In this work, the classifiers used in training were the Gradient Boosting Classifier, Random Forest Classifier, Logistic Regression, k-nearest Neighbors, and Decision Trees. Using those models was assessed on performance metrics to identify which can be used effectively for diagnosing Android ransomware in real-life situations.

5.3 Bayesian Game Model

First, the players and their strategies were defined to combine the developed machine-learning model with a Bayesian game model. The players included the attacker, who could either attack or not attack, and the defender, who could either use a machine learning-based defence or the typical defence. The determined machine learning model was utilised to predict the likelihood of an attack's occurrence. These probabilities fed into the defender's side by adding the likelihood of an attack into the defender's belief state.

[3] https://www.kaggle.com/datasets/subhajournal/android-ransomware-detection

Then, the payoff matrix was given to each strategy used by both parties involved in the strategic conflict. For example, if an attack was launched and the machine learning defence was activated, the defender would obtain a certain amount of profit, and the attacker would incur a certain amount of loss. However, other payoffs were assigned when a standard defence was applied or an attack did not occur. The probability that an attack would be launched shortly at the beginning period was assumed and transformed by the Bayesian inference approach into a posterior probability. This posterior probability integrated the likelihood of observing the current data given an attack together with prior beliefs about the likelihood of an attack.

The defender then used the posterior probability to develop the right strategic action. The necessity for using the machine learning defence increased when the updated probability of an attack was high. This decision-making process was based on reaching the Nash equilibrium, in which no player could gain a higher payoff by changing their strategy, knowing the other player's strategies.

Finally, the defender's chosen strategy was justified through modelling. These simulations analysed the strategy's success and compared the expected results with the real ones. In this sense, integrating the machine learning model with Bayesian game theory provided the corresponding dynamics to cybersecurity defence alongside probabilistic thinking to respond to potential attacks.

Table 1: Attributes For Bayesian Game Theocratic Model

Component	**Details**
Players	Attacker: Chooses between attacking or not attacking.
	Defender: Chooses between deploying ML defence or standard defence.
Strategies	Attacker: {Attack, do not attack}
	Defender: {Block, Monitor}
Payoffs	P_d1: Payoff for defender if blocked and attack occurs.
	P_a1: Payoff for attacker if blocked and attack occurs.
	P_d2: Payoff for a defender if monitor and attack occur.
	P_a2: Payoff for the attacker if monitor and attack occur.
	If no attack occurs, the payoff for both players is zero.
Initial Belief	Prior Probability of Attack: The defender's initial belief about the likelihood of an attack, denoted as
Observed Data	Likelihood: Probability of observing the data given an attack, denoted as (P(\text{Data}
Posterior Belief	Updated Probability: The defender's updated belief about the likelihood of an attack is calculated using Bayesian inference.
Nash Equilibria Analysis	Nash Equilibrium is the point at which neither the attacker nor the defender can improve their payoffs by unilaterally changing their strategies.
Simulation	Running various scenarios to validate the effectiveness of the chosen strategy.

5.3.1 Equilibrium Analysis

Based on the cybersecurity literature, Nash Equilibrium refers to a security situation in which the attacker and defender have chosen their strategies to the extent that neither can gain from unilaterally changing them. This concept is fundamental since it enables both parties to assess and analyse the other's activities to attain the most appropriate defensive position.

In the Bayesian Game Model, the Nash Equilibrium was first defined by establishing the attacker and defender's strategies and payoff structures.

Attacker's Strategies:

a) Attack
b) Do not attack.

Defender's Strategies:

a) Deploy ML defence.
b) Deploy standard defence.

The Nash Equilibrium is arrived at when every strategy maximises the player's payoff given their opponent's strategy. The payoff matrix for the Bayesian game model is as follows:

Table 2: Payoff Matrix for Bayesian model

Strategy	Symbolic Values (Defender, Attacker)
Attacker Attacks, Defender Blocks	(+d1, -a1)
Attacker Attacks, Defender Monitors	(-d1, +a1)
Attacker Does Nothing, Defender Blocks	(d2, +a1)
Attacker Does Nothing, Defender Monitors	(d1, +a1)

5.3.2 Calculating Equilibrium

They are calculated using posterior probabilities obtained through the Bayesian approach and represent a Nash Equilibrium. The posterior probability modifies the defender's belief regarding the occurrence of an attack contingent on the actual data received. This enables the defender to decide whether to employ ML or standard defence.

a) Initial Belief: The defender is assuming a prior probability of attack based on experience or data collected in the past.
b) Observed Data: When new data is observed, it is modified using Bayesian updates about the probability of an attack.
c) Posterior Belief: The new view of the likelihood of an attack feeds into the defender's plan.

Based on the posterior probability value, the defender evaluates the probabilities of the payoffs that correspond to the implementation of the ML defence as opposed to the standard defence. On the same note, the attacker determines if they are to attack or not depending on the actions most likely to be taken by the defender.

5.3.3 Achieving Equilibrium

Nash Equilibrium is reached when:

a) When deciding whether to employ ML or standard defence, the defender decides based on which of the two yields the defender the highest expected payoff, given the posterior probability of an attack.
b) From the defender's perception, the attacker chooses to attack based on the latter's expected payoff.
c) If an attack's posterior probability is high, the defender will go for the ML defence. Being aware of this, the attacker might deem the costs of attacking to be a higher price tag than the value of attaining the goal in question and, therefore, could refrain from it.
d) Conversely, if the posterior probability is small, the defender will choose standard defence, and the attacker might consider it more rewarding to attack.
e) None can unilaterally alter the strategy to increase their payoff in both scenarios, indicating the presence of Nash equilibrium.
f) This way, by adjusting their strategies to observed data and applying Bayesian inference, both the attacker and the defender can always be in perfect reciprocation, which is beneficial in any cybersecurity context.

6 Evaluation

6.1 Model Evaluation

Table 4 evaluates a Gradient-Boosting Classifier regarding various client-specified performance metrics. These statistics reveal that with a total support of 78,407, the classifier has a reasonably good accuracy rate of correct classification throughout the groups.

Table 4: Detailed Performance Metrics for Gradient Boosting Classifier

Attribute	Value
Accuracy	0.96
Macro Average Precision	0.97
Macro Average Recall	0.96
Macro Average F1 Score	0.96
Weighted Average Precision	0.96
Weighted Average Recall	0.96
Weighted Average F1 Score	0.96
Total Support	78407

6.2 Payoff Matrix Evaluation

The payoff matrix in Table 5 for the Gradient Boosting Machines (GBMs) included the attacker and defender's profile based on strategic interactions. The GBM model presented a consistent set of payoffs.

Table 5: Bayesian Pay-off Matrix for Trained Models

Model	Attacker Attack, Defender Block	Attacker Attack, Defender Monitor	Attacker Do Nothing, Defender Block	Attacker Do Nothing, Defender Monitor
GB Classifier	(+d1, -a1)	(-d1, +a1)	(d2, a1)	(d1, a1)

These observations suggested that the GBM model effectively prevented strong attack movements by defenders, leading to high defensive gains and attacker losses. Monitoring without blocking demonstrated overall neutral effects for defenders, with mixed outcomes for attackers depending on their actions. Identifying a constant target stimulated multiple defences to ensure the actualisation of attacks.

6.3 Expected Utility Calculation

Starting from the expected utility graphs depicted for both the attacker and the defender across the various models, the two strategies, namely, Strategy 1 and Strategy 2, relate to two different choices/actions that the two entities/cyber actors have at their disposal. In actuality, the applied principle for the first Strategy of the attacker is to attack. In contrast, for the second logic to the game of Strategy 1, the defender needs to counter the attack. Similarly, non-action or withdrawal is the action that the attacker has to take for Strategy 2, while the defender observes rather than blocks.

Table 6: Expected Utilities

Model	Attacker (S1)	Defender (S1)	Attacker (S2)	Defender (S2)
GB Classifier	0	0.2708	1.02708	-1.02708

These expected utility values represent (or capture) the outcomes or payoff of these strategies for the attacker and the defender in this research. For example, variables in the graph show that the Gradient Boosting Classifier provided predictions on the anticipations of the general utilities of these strategies and displays the effectiveness of various types of defensive and offence mechanisms. The bar graphs show these predefined strategies that are significant in using game theory in cybersecurity issues.

From the expected utility plot of the Gradient Boosting Classifier, we observe that once more, Strategy 1 carries positive utility for the defender (0.2708) and zero utility for the attacker. This implies that the defender blocks the attack, which means there is no gain on the attacker's side. Strategy 2 is favourable for the attacker with a utility equal to 1.02708 as opposed to the defender -1.02208. Although the attacker's utility is positive, the defender's utility means that the attack was, to some extent, counter-measured.

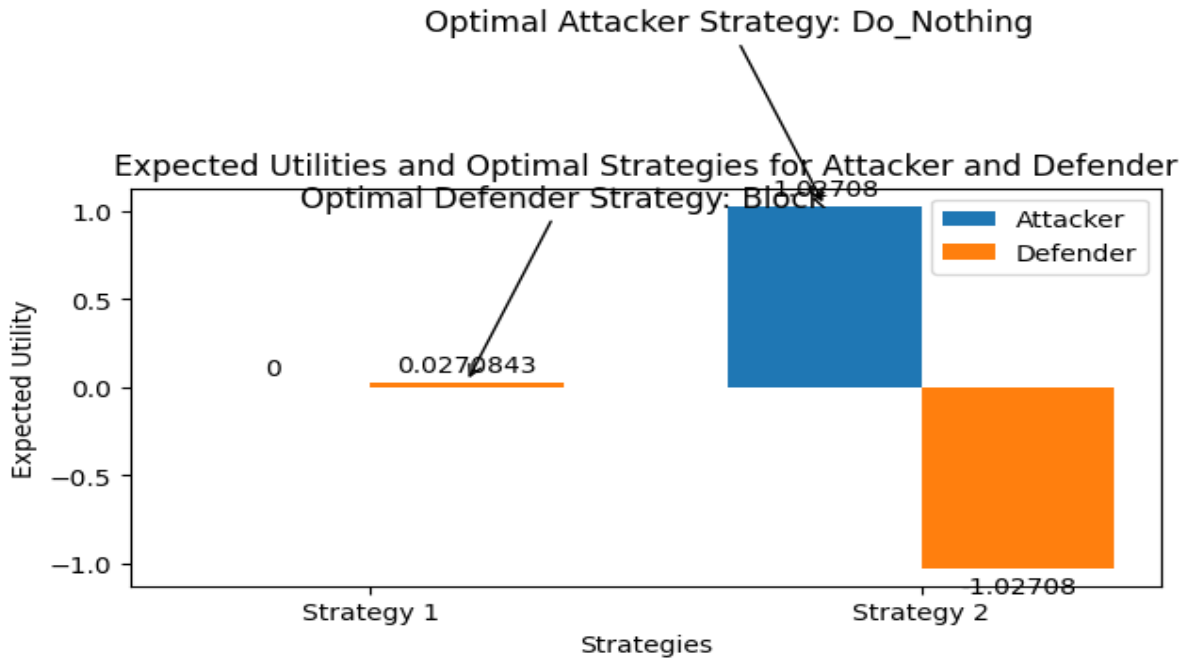

Figure 2: Expected Utilities and Optimal Strategies for Gradient Boosting Classifier.

All in all, the expected utility values indicate that the blocking strategies help promote the defender's positive results. At the same time, the attackers are bound to incur losses. Monitor, in turn, does not impact the defender's payoffs but implies a negative and stable pay-off for the attacker if he chooses to do nothing. This analysis shows how valuable the role of the blocker is when it comes to countering the attacks rather than the weakness of the attacker who does not do anything.

6.4 Prior & Posterior Beliefs

In Figure 3, the vertical axis shows predictions on the values from the dataset of two entities, attacker and defender, on the horizontal axis. There were several instances in which their evolution was showcased. This graph used the prior and posterior probability distributions about the success (**α**) and failure (**β**) rates of attacks.

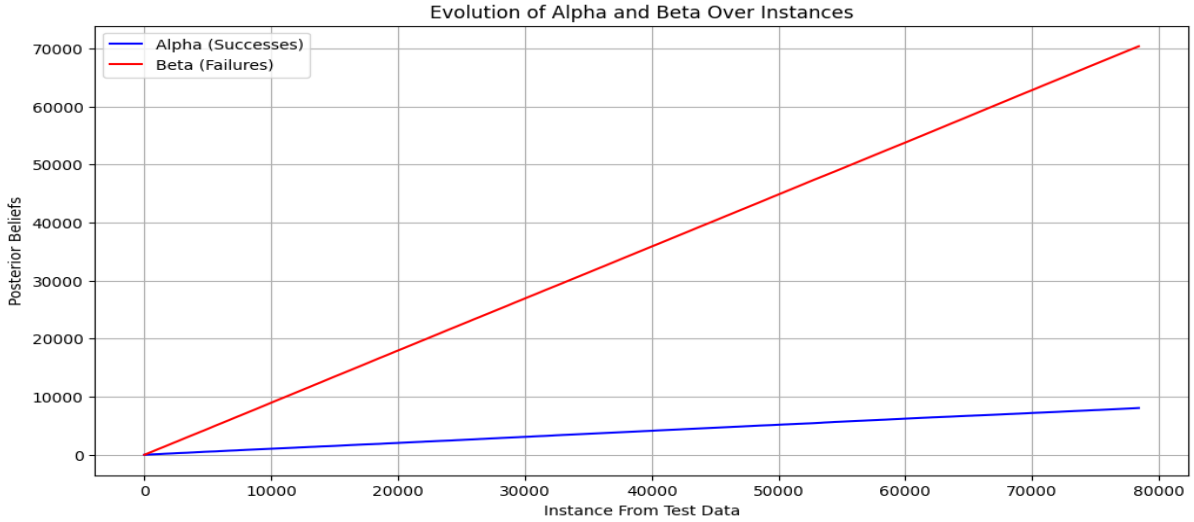

Figure 3: Evolution of α (Attacker success) and β (Attacker Failures) over instances.

It was a graph with predictions on the values from the dataset of two entities, the attacker and the defender, on the vertical axis, and several instances of their evolution showcased on the horizontal axis. This graph was created using the prior and posterior probability distributions about the success (Alpha) and failure (Beta) rates of attacks.

Starting with Alpha and Beta in the beta distribution, they were set to one, which is basic when handling uncertainty in the beta distribution. For each instance in the dataset, the probability of an attack's success was applied to the Beta distribution parameters. Specifically, the code determined the probability of success and failure for the instance at each stop and adjusted the Alpha and Beta values. Here, α1 represented the attacker's successes, shown by the blue line, and β1 represented the attacker's failures, shown by the red line. By common sense, these values indicated the defender's outcomes, with α2 for defender success and β2 for defender failure.

In the graph, the abscissa (X-axis) showed the number of instances while the ordinate (Y-axis) pointed to the values of Alpha1 and Beta1. The blue-coloured line for Alpha1 (attacker's successes) increased steadily as more instances were processed. In contrast, the red line for Beta1 (attacker's failures) was much steeper, indicating that the attacker encountered quick failures more than quick successes. This pattern demonstrated that attacker successes were far lower than attacker failures, implying that defender wins were far more frequent than defender failures.

This showed that with each evaluated case, the defender was in a more advantageous position to counteract the attacker's success. Using the Beta distribution to reflect ignorance and the adjustments made to these parameters with each new instance illustrated how the model's beliefs were updated based on the observed data, effectively capturing the dynamic nature of threats.

- **α1** = Attacker Success
- **α2** = Defender Success
- **β1** = Attacker failure
- **β2** = Defender failure
- **Instances**=Predictions on values from the dataset
- **Value**=Beliefs of Attacker & Defender on respective Instances

6.5 Closing remarks

The study outcomes indicated that the chosen algorithm, the Gradient Boosting Classifier, proved to classify and prevent cyber threats with great accuracy and efficacy. In particular, the accuracy of the classifier was up to 96%, and it demonstrated high values concerning characteristics such as precision, recall, and F1-measure. The matrix analysis pointed to the fact that the presence of the system reduced the likelihood of staging strong attack movements, an aspect that holds tremendous defensive advantages. The results of the expected utility calculations indicated an overall positive value for the defender and a negative value for the attacker with blocking strategies present. Furthermore, the use of Beta distribution for stressing prior and posterior beliefs revealed that the system can learn about threat probabilities. However, defenders claim success over attackers most of the time. Each outcome supported this hypothesis and reinforced that the optimal ML honeypot significantly enhances the system's security against existing and emerging cyber threats.

6.6 Discussion

In this study, three innovative techniques have successfully protected healthcare networks from ransomware attacks: machine learning, game theory, and honeypots. A gradient-boosting machine is suitable for big data with numerous predictors. It has a low risk of overfitting and is ideal for the dynamic nature of threats. The service provided the company with a means of quickly and correctly identifying threats that impacted the organisation, therefore enabling efficient protection against attacks by the IT team. Table 3 contains evaluation metrics for various classifiers, including the Gradient Boosting (GB) classifier, Decision Tree, Logistic Regression, Random Forest, and k-nearest Neighbors (k-NN).

Table 3: Performance Metrics for Trained Algorithms

Algorithm	Accuracy	Precision	Recall	F1 Score
Decision Trees	0.9857	0.9857	0.9857	0.9857
k-Nearest Neighbors	0.8056	0.8072	0.8056	0.8053
Logistic Regression	0.5450	0.5461	0.5450	0.5271
Random Forest Classifier	0.9403	0.9406	0.9403	0.9401
Gradient Boosting Classifier	0.9600	0.9600	0.9600	0.9600

According to these evaluated results, the Gradient Boosting Classifier model was the best choice for further application. While Decision Trees yielded slightly greater accuracy (98%), GBC was preferred over it for its better margin, ability to handle multivariate data, lesser chances of overfitting, and better generalisation ability. This makes it the most appropriate for diagnosing Android ransomware in real-life conditions. The basis for selecting the Gradient Boosting Classifier as the model of choice for implementation is rooted in the model's accuracy and reliability.

Using game theory, cyber attackers' strategies were analysed and reciprocated to minimise an organisation's vulnerabilities. The expected movements of opponents have been effectively studied, allowing the defenders to strengthen positions in the areas where attacks were most likely to be launched, thus increasing the extent of difficulty that the opponent would face in executing their plan. Therefore, the trained honeypots were doubly helpful in strengthening the security of a network. First, they mimicked the original target and, when attacked, were ready to provide what seemed or appeared to be valuable data. It also concealed the information, allowing the leakage to be traced and tracked.

The study by Wang et al. was primarily focused on increasing the security of Advanced Metering Infrastructure (AMI) in smart grid networks against DDoS attacks. AMI created a honeypot, a strategic game model implemented on AMI networks to monitor attacks. Honeypots were deployed in the systems, and the study in question deployed honeypots as decoy systems for the attackers. The model used Bayesian game theory to establish the plans for attack and defence, and it agreed with the assertions of multiple Bayesian Nash equilibria. Several experiments were performed on an AMI testbed, where it was found that by implementing the above-said strategy, the defence efficiency could be improved alongside

the reduction in energy consumption, and simultaneously, the detection rate of the DDoS attacks could also be increased. From this approach, it was comprehended that honeypots and game theory can be deployed as key strategies towards enhancing the security of smart grid networks against future cyber threats (Wang et al., 2017).

The most significant disparity between the two works is the area of concentration and techniques applied. The first study focused on protecting the AMI layer of the smart grid against DDoS attacks through an adaptive honeypot game model and the Bayesian game theory for analysing the defence strategies and improving the detection ratio and energy consumption (Wang et al., 2017). On the other hand, this model was centred on defending healthcare networks against ransomware attacks, with the use of Gradient Boosting Machines (GBMs) for threat identification, the use of honeypots for the analysis of attackers' behaviours as well as the application of Bayesian game theory in the adaptation of defence mechanisms with a high accuracy recording of 96%. The honeypots and the game theory used in both studies were similar, but the targeted uses and threats were different.

The study is highly accurate, with commendable features in detecting ransomware. It incorporates several advanced methods to form a flexible and efficient system. It is important to admit that the work of this research is solid and relatively close to perfect, but some aspects should be brought to attention. Results can be different depending on tested data sets or live examples. Its specificity on healthcare networks targets specific issues and solutions to preventing the leakage of patients' sensitive data to ransomware. However, there are potential limitations in applying the honeypot and machine learning because of the resource demands in constantly monitoring the honeypot and integrating Bayesian game theory in real-time. Altogether, the combination of machine learning, honeypots, and game theory can be concluded as a solid concept to improve the protection of healthcare networks from cyber threats.

7 Conclusions and Future Work

In this paper, the evaluation used Gradient Boosting Machines (GBMs) to determine the existence of ransomware by extracting its features from network traffic. The proposal to deploy honeypots was to implement them within the structure of the healthcare network to investigate ransomware attacks. Furthermore, the probabilistic threat approach used Bayesian game theory to adapt the network security architecture to real-time threats. In conclusion, this study revealed that using a composited approach strengthens the security of a computer network. These studies bring the possibility of integrating machine learning algorithms, Honeypot, and game theory to develop an effective and smart security system that minimises ransomware threats in healthcare networks. As for further work, it might be more suitable to refine these models and discover related fields where these models can be applied to increase the effectiveness of cybersecurity even more.

Future work could be done on implementing deep learning models for enhanced performance and creating flexible strategies to use when new threats are identified. At the same time, aspects of ethics concern more intricate approaches to cybersecurity. It is worth adding that collaborations with the industry's stakeholders seem like a key area that might further improve the model's usability. Future studies should extend the developed integrated models to other domains, such as insider threat detection or the protection of the Internet of Things (IoT). The enhancement of machine learning with Bayesian game theory's future potential is strong when considering the application in the commercial domain, including finance,

healthcare, and infrastructure. This paper proves that integrating these approaches in cybersecurity is a possible direction for future research that may lead to definite improvements in security systems.

References

Selvaraj, R., Kuthadi, V. M., & Marwala, T. (2016). Honey pot: A major technique for intrusion detection. In Proceedings of the Second International Conference on Computing and Communication Technologies (pp. 73-81).

Yeldi, S., Gupta, S., Ganacharya, T., Doshi, S., Bahirat, D., Ingle, R., & Roychowdhary, A. (2016). Enhancing network intrusion detection system with honeypot. Pune Institute of Computer Technology.

Thakar, U. (2005). HoneyAnalyzer: Analysis and extraction of intrusion detection patterns & signatures using honeypot. In Proceedings of the Second International Conference on Innovations in Information Technology.

Khurana, S. (2023). Ransomware threat detection and mitigation using machine learning models. In Proceedings of the IEEE International Conference on ICT in Business Industry & Government (ICTBIG), Indore, India.

Ahmad, S., Zulkifli, Z., Nasarudin, N. H., Imran, M., & Ariff, M. (2023). A recent systematic review of ransomware attack detection in machine learning techniques. In Proceedings of the 4th International Conference on Artificial Intelligence and Data Sciences (AiDAS), IPOH, Malaysia.

Alraizza, A., & Algarni, A. (2023). Ransomware detection using machine learning: A survey. Big Data and Cognitive Computing, 7(3), 143.

Selvaraj, R., Kuthadi, V. M., & Marwala, T. (2019). A game theoretical based system using Holt-Winters and genetic algorithm with fuzzy logic for DoS/DDoS mitigation on SDN networks.

Patel, R., Shah, K., Trivedi, N., & Amin, P. (2020). Reputation management using honeypots for network security.

Krishnaveni, S., Prabakaran, S., & Sivamohan, S. (2018). A survey on honeypot and honeynet systems for intrusion detection in cloud environment. Journal of Computational and Theoretical Nanoscience, 15(9/10), 2949-2953.

Han, W., Yang, L., Li, X., & Li, Y. (2016). Honey mix: Toward SDN-based intelligent honeynet.

Jiang, X., Xu, D., Wang, Y., & Jia, Y. (2011). Design and implementation of dynamic virtual network honeypots.

Nawrocki, M., Dandurand, L., & Adhikari, K. (2016). A survey on honeypot software and data analysis.

Roychowdhary, A., Gupta, S., Yeldi, S., & Bahirat, D. (2020). Collaborative honeypot defense in UAV networks: A learning-based game approach.

Kyung, S., Kim, J., Park, Y., & Choi, S. (2017). HoneyProxy: Design and implementation of next-generation honeynet via SDN.

Naeem, A. A. N. (2021). Honeypots: Concepts, approaches and challenges. Retrieved from https://hal.science/hal-03324407

Bedi, H. S., Roy, S., & Shiva, S. (2011). Game theory-based defense mechanisms against DDoS attacks on TCP/TCP-friendly flows. In IEEE Symposium on Computational Intelligence in Cyber Security (CICS) (pp. 129-136). IEEE.

Wang, K., Du, M., Maharjan, S., & Sun, Y. (2017). Strategic honeypot game model for distributed denial of service attacks in the smart grid. IEEE Transactions on Smart Grid, 8(5), 2474-2482.

Prabowo, W. A., Fauziah, K., Nahrowi, A. S., Faiz, M. N., & Muhammad, A. W. (2023). Strengthening network security: Evaluation of intrusion detection and prevention systems tools in networking systems. International Journal of Advanced Computer Science and Applications, 14(9), 107-115.

Franco, J., Aris, A., Canberk, B., & Uluagac, A. S. (2021). A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems. IEEE Communications Surveys & Tutorials, 23(4), 2351-2383.

Zhu, M., Anwar, A. H., Wan, Z., Cho, J. H., Kamhoua, C. A., & Singh, M. P. (2021). A survey of defensive deception: Approaches using game theory and machine learning. IEEE Communications Surveys & Tutorials, 23(4), 2460-2493.