DOI: https://doi.org/10.33103/uot.ijccce.24.2.3

Ransomware Detection and Prevention Using Machine Learning and Honeypots: A Short Review

Shayma Jawad[1], Hanaa Mohsin Ahmed[2],

[1, 2]*Department of Computer Sciences, University of Technology, Baghdad, Iraq*

[1]*cs.21.04@grad.uotechnology.edu.iq,* [2]*Hanaa.M.Ahmed@uotechnology.edu.iq*

Abstract*— Ransomware is a type of computer malware that is currently widespread and highly dangerous. Ransomware attacks have become a major cybersecurity risk, posing significant risks to individuals and organizations alike. Also, traditional techniques for malware analysis, such as signature matching and heuristics, are no longer viable due to the exponential growth of malware. Researchers have explored various approaches to address this issue, but there is a lack of proper documentation and comparison of existing works. This paper presents an analysis of ransomware detection systems, which is part of an ongoing research project aiming to develop an open-source ransomware detection system to address the identified gaps in the field. To prevent such attacks, it is recommended to regularly backup files and avoid clicking on untrusted email links and attachments. Machine learning has been suggested by researchers as a more effective method of detecting malware. With internet use on the rise, honeypots offer a viable option for reducing security threats and safeguarding classified data from hackers. Honeypots are regarded as valuable resources for thwarting attacks and giving important insight about the origin and behavior of such attacks, which is useful for analysts who conduct such investigations. This paper provides a general view of cybersecurity, machine learning (ML), cyber threats, and honeypot systems towards mitigating system attacks and understanding their origin and behavior.*

Index Terms*— Cybersecurity, Ransomware, Honeypots, Machine Learning, Detection.*

I. INTRODUCTION

The spread of ransomware over the past few years has turned it into an unmanageable cyber menace and an extremely lucrative criminal enterprise. Deep learning (DL) methods that use deep neural networks (DNNs) have gained popularity as high-performance computer resources have proliferated. Because it can deal with a lot of features while working with unstructured data, DL is able to operate with greater power and flexibility [1], [2].

Hyperphysical systems and the ICT sector are both in high danger from ransomware, a particular sort of cyberattack. It utilizes various encryption techniques to encrypt critical files on the victim's computer and can be decrypted using ML tools such as logistic regression, support vector machines (SVMs), decision trees (DTs), random forests (RFs), and DL algorithms like DNN, convolutional neural networks (CNN), and long short-term memory (LSTM). Through backpropagation and stochastic gradient descent optimization, DL algorithms seek to identify intricate patterns in data[3]. There are several modern technologies that can be used to detect malware [4], including:

A. Sandboxing or code emulators: This is a very strong tool against just-hatched malware. Such an application provides an opportunity for them to analyze the behavior of an object in a virtual environment and make smart decisions about whether that particular object is dangerous or not.

B. ML and AI: These tools can analyze lots of data and reveal clues about possible forthcoming assaults. They are useful for enhancing highly sensitive and proactive early warning systems.

Received 21/October/2023; Accepted 02/January/2024

ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.24.2.3

C. Heuristic analysis: It also involves analyzing the behavior of an item to find out if it is harmful or not. Malware that has never been detected before can be detected through heuristic analysis.

D. Memory execution monitoring: This involves monitoring the execution of code in memory to detect malicious behavior. This can be used to detect malware that is designed to evade traditional detection methods.

E. Tampering analysis technologies: These technologies can detect attempts to modify or tamper with system files or configurations. They can be used to detect malware that attempts to modify system settings to evade detection.

Researchers have suggested the use of honeypots, which are devices that have been purposefully placed and are intended to draw potential attackers and collect data on their actions. So as to improve the precision of detection, honeypots can help update ML models by supplying useful training data. It is vital to review pertinent literature in order to spot trends and obtain knowledge for potential future developments because putting honeypots into practice might be difficult in the absence of a defined plan [5].

Honeypots are computer resources designed to be probed, attacked, or compromised, providing valuable insights into attackers' actions and motivations. Honeypot research involves developing and deploying honeypot software as well as analyzing collected log data in a structured manner. These decoys complement traditional detection mechanisms and offer valuable insights into cyber threats [6].

In this paper, we will discuss the challenges posed by the dynamic nature of ransomware and the strategies that ML employs to adapt to evolving attack techniques through the use of honeypots. We will also provide a comprehensive overview of honeypot and methodologies for analyzing honeypot data. This review paper will cover various specific topics.

II. RANSOMWARE ATTACK

Ransomware is a sort of malware that keeps users from getting to their framework or their own records by encrypting them with a password, demanding payment in exchange for the password, and granting access to the files once again [7]. A type of malware known as ransomware encrypts the system or files and demands payment before unlocking them [7].

Ransomware, often known as "Ransom-Malware," is a type of malware that keeps victims from regaining access to their computers or personal files by encrypting them and then demands a ransom payment in exchange for the victim's release from the lockdown. Instructions are given to users on how to purchase the "Decryption Key" by paying money [8].

Ransomware is characterized into two sections relying on the pre-owned lock method [9]:

A. ***Computer's locker*****:** This section terminates the victim system processes by overloading system assets, while the information is unaffected.

B. ***Crypto-virus*****:** Utilize a high-level encoding technique to encode the desired information.

Ransomware's Impact [10]:

- Significant confidentiality data loss.
- Disruption of the systems' typical operation.
- The expense of recovering the data and paying the ransom.
- A threat to the dependability of the organization.

Probably a complete stop to organizational advancement.

Received 21/October/2023; Accepted 02/January/2024

DOI: https://doi.org/10.33103/uot.ijccce.24.2.3

III. LITERATURE REVIEW BASED ON RANSOMWARE DETECTION TECHNIQUES

The many detection methods used to find and recognize ransomware are covered in this section.

A. Review Based on Machine Learning Techniques

Malware is being detected with the help of ML methods. In static detection, applications are tested for malware without being executed, which is a subfield of malware detection. On the other hand, in dynamic detection, the software or applications are tested by running them. Hybrid detection employs both static and dynamic methods. On the author's custom data, the DT had the highest accuracy of 99.90 percent. SVM, on the other hand, outperformed DTs in terms of accuracy when applied to a malware dataset. SVM had the highest recall value, at 100%[11].

Such as ML and AI, can help mitigate the impact of cyberattacks. These systems can analyze a lot of data and find patterns that could point to an attack. Additionally, the paper notes that ransomware attacks often involve a series of steps, and detecting and preventing these assaults requires a diverse methodology that includes both specialized and non-specialized controls[12].

The fact that signature-based malware detection approaches are insufficient to combat malware since they may be readily tricked in an intelligent way is one of the difficulties in identifying malware using standard ML techniques. In addition, ML and DL algorithms need a significant quantity of training data to get started, and in malware and threat detection, one of the main obstacles is providing the algorithm with enough examples of malicious and benign content. Another significant obstacle is the potential for inaccurate findings due to "overtraining the model" caused by incorrectly classified and noisy data [13].

High-level features may be learned by DL algorithms from the data, and they often do not need hard-core feature extraction or domain knowledge. In comparison to conventional learning algorithms, these algorithms can provide findings that are more accurate when trained on vast volumes of data. In the majority of situations, DL models outperformed conventional learning techniques. Shallow learning approaches may not result in a scalable solution with substantial accuracy, while DL methods are capable of identifying and hunting malware or other threats.

Table I presents various ML-based methods for detecting malware on Android and Windows systems. The results demonstrate the effectiveness of ML in enhancing malware detection and improving security measures on both Android and Windows platforms.

TABLE I. ML-BASED MALWARE DETECTION METHODS FOR ANDROID AND WINDOWS

Ref, year	**Dataset**	**ML Technique**	**Type of Malware**	**Result**	**Weak Points**
[14], 2019	Ransomware dataset	ML Techniques: (SVM, RF, Naive Bayes)	Windows 10	SVM and RF accuracy of 99.5%, and the NB accuracy of 96%	It has several weak points, including limited ML techniques, a lack of diverse dataset, no comparison with other methods, etc.
[15] 2019	Ransomware dataset	"ML models: (Logistic regression, linear SVM, DT, RF, gradient boosting tree, SVM, MLP).	-	Different detection rates, from a minimum of 91% to a maximum of 99%, were attained.	The weak points, including a lack of detailed methodology, incomplete evaluation metrics, a lack of comparison with other methods, limited dataset description, and a lack of discussion on model interpretability.
[16] ,2020	VisDroid dataset that has five types of images.	ML (RNN and Inception3)	Android	accuracy reached 98.2%	The Weak Points, Including A Limited Dataset, A Lack Of Comparison With Other Methods, And A Lack Of Explanation On Model Selection.

Received 21/October/2023; Accepted 02/January/2024

DOI: https://doi.org/10.33103/uot.ijccce.24.2.3

[17], 2021	normal samples and malware samples	DNN, CNN, and (LSTM) recurrent neural network (RNN)	Windows 10 OS	ACC achieve 97% AUC achieve 98% F1-score with a far of under 1.88% on average	The weak points, including a lack of detailed dataset description, incomplete evaluation metrics, a lack of comparison with other methods, and insufficient explanation of model architectures.
[18] ,2021	publicly available dataset	Mybrid DL enabled intelligent multi vector	Android	demonstrates the performance in terms of detection accuracy and time efficiency.	The weak points, including a lack of detailed dataset description, incomplete explanation of the DL-based model, a lack of comparison with other methods, limited evaluation metrics, and insufficient discussion on model robustness.
[19], 2022	43867 individual snippets of data from assorted sources	DNNs that used embedded CNN layers CNN-LSTM	Windows	CNN-LSTM outperforms with 99% detection accuracy, 99% precision, and 99% recall.	The weak points, including a lack of detailed dataset description, insufficient explanation of the CNN-LSTM model, limited evaluation metrics, a lack of comparison with other methods, and limited discussion on real-time performance.
[20] ,2022	dataset of malware and benign apps	Deep-Layer Clustering	Android	The findings indicate that the classifier's performance is enhanced by the identified patterns.	The weak points, including a lack of detailed dataset description, insufficient explanation of the Deep-Layer Clustering method, limited evaluation metrics, a lack of comparison with other methods, and limited discussion on practical implications.
[21], 2022	Ransomware dataset	different types of neural networks (ANN, CNN, RNN)	-	accuracy rate 100%, outperforming other models like ANN, CNN, and RNN at 91%, 94%, and 79% respectively.	The weak points, including a lack of detailed dataset description, insufficient explanation of the neural network architectures, a lack of comparison with other methods, limited evaluation metrics, and a lack of discussion on model interpretability.
[22], 2022	ransomware and goodware instances.	DL based CSPE-R	Windows	**Method** | **Acc** | **Recall** | **F1** SVM | 0.88 | 0.79 | 0.87 RF | 0.80 | 0.76 | 0.79 LR | 0.90 | 0.81 | 0.89	The weak points, including a lack of detailed dataset description, insufficient comparison with other methods,and a lack of explanation of the CSPE-R method.
[23], 2022	70:30 ransomware and legitimate observations.	ML classifiers and NN-based architecture detect ransomware using traditional methods.		achieved highest mean AUC (0.99) scores	The weak points, including limited dataset description, insufficient explanation of the ML classifiers and NN-based architecture, a lack of comparison with other methods, limited evaluation metrics, and insufficient discussion on real-world applicability.
[24], 2022	Web-Crawler (‘GetRansomware’)	ML algorithms (LR, SGD, KNN, NB, RF, SVM)	Windows	Achieved accuracy of 99.15%	the weak points, including limited dataset description, insufficient explanation of the ML algorithms, a lack of comparison with other methods, limited evaluation metrics, and insufficient discussion on explainability.
[25], 2022	ransomware samples and goodware samples.	ML models are DT, RF, KNN, SVM, XGBoost and LR.	Windows	**name** | **precision** | **recall** DT | 0.92 | 0.97 RF | 0.92 | 0.98 KNN | 0.89 | 0.95 SVM | 0.93 | 0.97	The weak points, including a lack of detailed dataset description, insufficient explanation of the ML models, a lack of comparison with other methods, limited evaluation metrics, and a lack of

Received 21/October/2023; Accepted 02/January/2024

DOI: https://doi.org/10.33103/uot.ijccce.24.2.3

				XGB 0.96 0.99 LR 0.97 0.98	discussion on limitations and future directions.
[26], 2022	staff office users accessing shared files uninfected.	Neural network model (NN)	Windows	Identified 100% of 10 unutilized crypto-ransomware binaries with 99 MB data loss.	The weak points, including a limited scope of the study, lack of detailed methodology description, insufficient evaluation metrics, a lack of comparison with other methods, limited data loss analysis, and a lack of discussion on limitations and future directions.
[27] ,2022	authoritative dataset Drebin	Multi-granularity opcode features, TFIDF algorithm, DL model, Resnet	Android	malware detection accuracy is 96.35%, obfuscated malware detection accuracy is 94.55%.	The weak points, including limited dataset description, insufficient explanation of the multi-granularity opcode features, lack of detailed methodology description, limited evaluation metrics, a lack of comparison with other methods, and a lack of discussion on limitations and future directions.

As shown in Table I**,** a different technique has been developed to find Windows malware. One method entails the implementation of a malware detection framework employing active learning, where a SVM classifier is used to categorize anonymous malware in the Windows operating system environment as either legitimate or harmful. In order to strengthen security measures, this approach aids in the identification and differentiation of various forms of malware. As for Android, ML methods are utilized to assess the performance of feature vectors acquired from audio signals, and the feature set for this detection strategy is produced from audio signals retrieved from executable files to find malware on Android devices.

B. Review Based on Honeypots

A honeypot is a security resource that is purposefully created to be investigated, attacked, or hacked in order to acquire data about attack patterns, hacker motivations, and regularly used applications. Because it does not stop specific incursions or viruses, it is more of a detection and reaction tool than a preventive strategy. Honeypot data assists defenders in strengthening their defenses and developing countermeasures against future security threats, while also offering insights into hackers' technical skills [28].

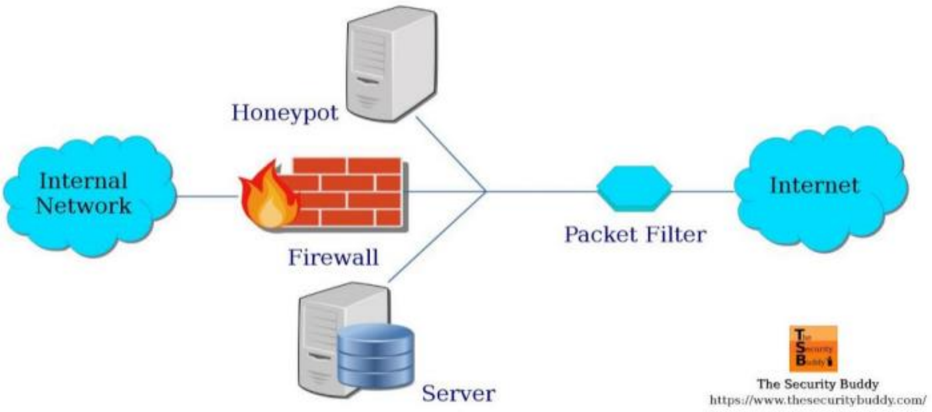

FIG. 1. HONEYPOT FOR IMPROVING SECURITY [28].

Honeypots are cybersecurity technologies that may be used to catch and redirect intruders away from real-world production systems as depicted in *Fig. 1*, are security tools without any actual or production value. They are most successful when controlled by people who understand security, systems, and networks. Honeypots, on the other hand, can be abused by hostile actors and may not

Received 21/October/2023; Accepted 02/January/2024

DOI: https://doi.org/10.33103/uot.ijccce.24.2.3

appreciably improve network security if managed incorrectly [28]. An information security policy's primary goal is to guarantee the availability, accessibility, safety, and authenticity of services. The peculiarity of the honeypot resides in the fact that it publicly presents itself as a vulnerable system likely to draw the attention of hackers. Attacks rely on programs that scan a network, seeking flaws [29].

To counter such a threat, several scholars have lately started concentrating on how to automatically identify a honeypot server. Researchers must make their honeypot services more realistic while also enhancing both the internal workings and the exterior user interface.

Haltaş, F. et al. [30] the researchers have suggested a number of detection systems that use host-monitoring methods and ML algorithms to evaluate malware samples so as to improve the security of IoT devices. However, the deployment of these methods in actual corporate networks and the requirement for a considerable amount of storage capacity frequently present difficulties. Additionally, malware samples are frequently gathered using honeypot systems for analysis and spotting prospective assaults.

H. T. Nguyen, et al. [31], They introduce the concept of IoT botnet detection, highlights the potential of CNN-based approaches, and presents the results of the proposed PSI graph CNN classifier, demonstrating its effectiveness in identifying Linux IoT botnet malware. The results indicate that the PSI graph CNN classifier achieved a high accuracy of 92% and an impressive F-measure of 94%.

Matin, and Rahardjo B. [32], They discuss the way malware is becoming a bigger menace and how old-fashioned security measures can't keep up. It suggests an alternate method for detecting malware by categorizing distinct types: honeypots paired with ML techniques, particularly DT and SVM. The article discusses the experimental approach to be employed together with an architectural design solution.

O. P. Dwyer, et al. [33], The article introduces a new method for identifying Mirai-like botnet activities using DNS-based data from real datasets. It demonstrates how using a single DNS record can improve botnet profiling for IoT devices. The method significantly reduces botnet detection time and achieves an average accuracy of 99% using the RF formulation. The study also employs multiple ML classifiers.

Vishwakarma R., and Jain A. K., [34], They suggest a technique for detecting malware on IoT devices that combines honeypots with ML. They build an ML model efficiently and dynamically using the IoT honeypot data as a dataset. This strategy tries to tackle the problem of protecting IoT against zero-day DDoS attacks, which have grown to be a major concern in IoT security.

Alhaidari S., & Zohdy M. [35] In their research, a hybrid learning technique is suggested to address the problem of high false alarms and middling accuracy in IoT intrusion detection systems. This strategy combines Hidden Markov Model (HMM) technology with partitioning and clustering approaches. The use of K-Medoids in the suggested technique increases detection rates and lowers false-positive rates, according to experimental findings.

Tien C. et al. [36] a methodology for efficiently analyzing malware aimed at IoT devices is introduced. This framework uses ML to identify and categorize the malware into recognized categories after extracting universal characteristics from it using static analysis. A dataset of more than 6,000 IoT malware samples was used to assess the framework, which showed great accuracy in identifying and categorizing malware that is a danger to IoT devices.

Shobana M. and Poonkuzhali S. [36] Their model detects IoT malware by analyzing system calls during execution. The Strace tool in Ubuntu collects these calls, preprocesses them using n-gram techniques, and classifies them into normal and malicious using an RNN. The effectiveness of this DL approach is evaluated using performance metrics, and real-time IoT malware samples are collected from the IOTPOT honeypot.

Malware may be categorized according to shape. This has an impact on the malware targets. Different kinds of malware can influence the features employed from an ML standpoint. The

Received 21/October/2023; Accepted 02/January/2024

DOI: https://doi.org/10.33103/uot.ijccce.24.2.3

performance of the model is substantially determined by these factors. According to the study's findings, there are three different forms of malware: internet-connected device malware, portable Windows executable malware, and malware streams [38]. Malware types are presented in Table II.

- Malware that targets **Internet of Things (IoT)** architectures, including ARM, M68K, SPARC, MIPS, PPC, SH4, and others, is referred to as IoT-based malware. IoT devices and networks are at risk from this virus, which is divided into families like ZORRO, GAYFGT, nttpd, KOS, and *.sh. This malware may jeopardize the security and functionality of IoT devices and networks [39].
- **A Portable Executable (PE)** file is a file format for executable files in Windows operating systems. It includes critical headers such as the PE file header and optional headers that offer information about the file's compatibility, computer requirements, number of pieces, and other characteristics. The optional header, in particular, determines the logical structure of the PE file and contains important elements such as the entry address, OS version, and subsystem version. It is critical to understand these headers when evaluating and working with PE files in ML applications [40].
- **A Data Stream** is a type of data format used to record and store data about network activity. A honeypot data stream keeps track of a variety of information, including IP addresses, protocols, services, packet contents, and packet destinations. The analysis of network behavior and the discovery of possible risks or anomalies benefit greatly from this knowledge [38].

The collection of information on honeypots is critical to ensuring the availability of data required by ML while executing training models. Several researchers received a malware collection directly utilizing a virtual honeypot during the operation. Researchers may utilize data directly in this manner without needing to construct a honeypot architecture, as shown in Table II. As shown in Table II, various techniques and their corresponding results for different types of malware and honeypots are presented. These results provide insights into the effectiveness of virtual honeypots and open-source honeypot datasets in detecting and analyzing attacks in the IoT domain.

TABLE II. TECHNIQUES FOR HONEYPOT

Ref, Year	Type of Malware	Honeypot Techniques	Result
[30], 2014	Malware streams	Virtual Honeypot	NCD was 0.5375
[31], 2018	IoT	Open sources honeypot	ACC was 92% and F-measure was 94%.
[32], 2019	Portable Executable (PE)	Virtual Honeypot	-
[33], 2019	IoT	Virtual Honeypot	ACC was 99%
[34], 2019	IoT	Virtual Honeypot	-
[35], 2019	IoT	Open sources honeypot dataset	ACC was 0.9467
[36], 2020	IoT	Open sources honeypot dataset	ACC was 99% and F1-score was 97%
[37], 2020	IoT	Open sources honeypot dataset	ACC was 98.712 %

- **A Virtual Honeypot:** is a trap system built with virtualization technologies that can run on a single machine [34]. It is capable of simulating network traffic and acting like a genuine system. Because of the ease of installation using VMWare, User Mode Linux, and Microsoft Virtual PC,

Received 21/October/2023; Accepted 02/January/2024

DOI: https://doi.org/10.33103/uot.ijccce.24.2.3

virtual honeypots are widely employed. They provide benefits such as simple separation and repair, as well as the ability to simulate many systems on a single machine [41].

- **The Open-Source Honeypot Dataset:** This dataset contains 6631 suspicious benign and harmful flows that were gathered in the last quarter of 2007 from two honeytrap honeypot instances. As well as different dangerous actions like buffer overflows, exploits, shellcodes, and viruses, it also covers seemingly innocent yet dubious operations like HTTP, FTP, and database connection requests [38].

IV. CLASSIFICATIONS OF HONEYPOTS

Two classifications are used for classifying honeypots: production honeypots and research honeypots [42].

A. Production Honeypots: are designed for easy deployment and use in company production environments to enhance network security by diverting attacks. However, there is a trade-off between the simplicity of operating these honeypots and the amount of information they can collect about the attacks. In other words, while production honeypots are convenient to manage, they may provide limited insights into the nature and details of the attacks.

B. Research Honeypots: Research honeypots are a type of honeypot that provides detailed information about attacks but is more challenging to set up. They are commonly utilized by research organizations and network forensics scientists to analyze attacks, develop countermeasures, and gain insights into the motives, behavior, tools, and structure of malicious actors in the black-hat community.

V. HONEYPOT'S CLASSIFICATION BASED ON THE LEVEL OF INTERACTION

Honeypots, which are computer systems designed to appear normal to attackers while logging their actions, can be classified based on the attacker's level of participation, the type of data gathered, and the setup of the system [43]. The level of interaction classification focuses on the extent to which the attacker can interact with the honeypot, with high-interaction honeypots allowing full interaction and low-interaction honeypots providing limited functionality to detect unauthorized activity [44]. The more interactive the honeypot, the closer it resembles real targets, potentially yielding more accurate information about attacker techniques [45].

A. Low-interaction honeypots (LIHP) are a type of honeypot that mimics only a few interactions, like SSH or FTP, and denies the attacker access to their OS. They have a low response, which is used for protocol handshake and just provides information for statistical analysis. They cannot collect a lot of information, but they can recognize peaks with request numbers when there is an attack by autonomous worms or they work in a production environment with large data traffic.

B. Medium-interaction honeypots (MIHP) provide higher interaction, although they do not emulate OS. These MIHPs provide an advanced interaction to lure attackers and possibly generate further attacks. Yet, because of reduced interaction with external users for both LIHP and MIHP, they will be less prone to attack.

C. High-interaction honeypots (HIHP), a sophisticated and complex honeypot in which attackers encounter a real environment with an operating system and many services, have been discussed. They acquire comprehensive details like attack logging, data access, file traversal, and running code. The logs analyzed in cases of high-interaction honeypots are usually performed by network forensic experts who work very intensively and mainly manually. These are the types of honeypots that researchers often employ.

While there is a classification of honeypots based on the level of interaction, it is considered academic and impractical due to the diverse range of honeypot variations. Instead, it has become common to distinguish between low- and high-interaction honeypots, where low-interaction honeypots

Received 21/October/2023; Accepted 02/January/2024

DOI: https://doi.org/10.33103/uot.ijccce.24.2.3

are limited to port listeners or service emulators, while high-interaction honeypots provide real services and aspects of an operating system. This classification simplifies the categorization of honeypots, as recommended by Lance Spitzner [42], [46].

VI. COMPARISON BETWEEN RANSOMWARE DETECTION METHODS

Honeypot and ML are two different approaches used for ransomware detection, each with its own strengths and limitations.

In this section, we focused on the comparison between techniques used for ransomware detection. The comparison is summarized in Table III.

TABLE III. COMPARISON BETWEEN HONEYPOT VS ML DETECTION

Division	Honeypot	Machine Learning
Advantages	The honeypot technique involves setting up decoy files that wait to be attacked by ransomware. It has an advantage since the system needs less maintenance and has little processing power. Nonetheless, we cannot guarantee that ransomware will aim at those honeypot files, so it is also important to notice the characteristics of the most probable victim files.	• When trained with a mixture of training data representative of different outcomes, ML can successfully make predictions concerning those outcomes. • The reason behind using ML in ransomware detection is that ML learns patterns from the data and thus becomes less prone to obfuscation methods.
Disadvantages	This method is not foolproof; the ransomware may or may not attack decoys called honeypot files. As such, it is necessary to define specific features of the likely target files, which will increase the efficiency of the honeypot method.	• Not all the times you use the right algorithms for ML, and the process requires lots of iterations. • Caution should be taken in order to avoid bias or overfitting, which may result in a lack of the correctness model’s accuracy and generalization.
Research Gap	A lack of attack makes a honeypot have a few limitations and can also be inaccurate with information, making the information lack depth. Improved effectiveness and reliability will depend on understanding the specific characteristics of files likely to be targeted.	The research gap in ML for ransomware detection is significant, as no robust models or solutions have been developed to effectively protect against such attacks, highlighting the need for further development.

As show in Table III, the honeypots are decoy systems that mimic real targets to attract attackers, observing ransomware behavior. They offer insights into attacker techniques but are limited in detecting all types of attacks. ML algorithms analyze large amounts of data to identify malicious activities associated with ransomware, learning from historical data and adapting to new threats. ML has shown promise in detecting previously unseen malware variants. In terms of comparison, we can briefly express it using the information in Table III:

1. Honeypots rely on attracting actual attacks while ML uses historical data for analysis.

2. Honeypots offer insight into attacker behavior but may not always be effective at capturing all types of attacks.

3. ML algorithms have the potential to detect unknown or evolving forms of ransomware based on learned patterns.

Both approaches have their limitations: honeypots may not always attract attackers, while ML algorithms require continuous training updates as new threats arise.

VII. CONCLUSIONS

This paper examines the use of ML and honeypots for malware detection, highlighting their potential to overcome the limitations of traditional methods. It discusses recent techniques, such as honeypots and ML models, focusing on detecting trending malwares like ransomware, Windows, and Android. While ML-based systems are more accurate in detecting

Received 21/October/2023; Accepted 02/January/2024

DOI: https://doi.org/10.33103/uot.ijccce.24.2.3

ransomware attacks, they have a slower detection time and cannot completely prevent attacks, leaving encryption risks.

Based on the information provided in Table I, it is difficult to determine which method is the best as it depends on various factors, such as the specific type of malware being detected and the platform (Android or Windows) being used. Each method has achieved high detection accuracy, ranging from 91% to 99.5%. It would be best to consider other factors, such as implementation feasibility, computational resources required, and any specific requirements for your particular use case, when determining which method may be most suitable for your needs. Based on the information provided in Table II, it is difficult to determine which technique or honeypot is the best for detecting and analyzing attacks in the IoT domain. Each technique has different results and may be more effective for specific types of malware or scenarios. It would require further analysis and comparison to determine which technique or honeypot is considered the best in this context. As for Table III, it provides a comparison between honeypot and ML techniques, highlighting their advantages and disadvantages. The effectiveness of each technique may vary depending on the specific context and requirements of the system being protected against ransomware attacks.

Alternatives like honeypots and network analysis show more effectiveness. Honeypots can be used to improve existing honeypots or develop new security systems in the future by capturing network intruders and learning about their methods. The goal is to develop a more accurate and efficient ransomware detection system.

REFERENCES

[1] R. A. Alsaidi, W. M. Yafooz, H. Alolofi, G. A. M. A. H. M. Taufiq-Hail, Emara, & A. Abdel-Wahab, "Ransomware Detection using Machine and Deep Learning Approaches.," *International Journal of Advanced Computer Science and Applications,* vol. 13, no. 11, 2022.

[2] A.Mathew, P.Amudha, & S.Sivakumari, "Deep learning techniques: an overview.," *Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020,* pp. 599-608, 2021.

[3] N.Rani, and S. V. Dhavale, "Leveraging machine learning for ransomware detection," *arXiv preprint arXiv:2206.01919.,* 2022.

[4] L. Amer, "Malware development threats with modern technologies," *Cyber Security: A Peer-Reviewed Journal,* vol. 6, no. 2, pp. 178-187, 2022.

[5] I. M. M. Matin, & B.Rahardjo, "The use of honeypot in machine learning based on malware detection: A review.," *In 2020 8th International Conference on Cyber and IT Service Management (CITSM), IEEE.,* pp. 1-6, (2020, October).

[6] E. Vasilomanolakis, S. Karuppayah, P. Kikiras, & M. Mühlhäuser, "A honeypot-driven cyber incident monitor: lessons learned and steps ahead.," *In Proceedings of the 8th International Conference on Security of Information and Networks,* pp. 158-164, 2015.

[7] N. Alzahrani, & D. Alghazzawi, "A review on android ransomware detection using deep learning techniques.," *In Proceedings of the 11th international conference on management of digital EcoSystems,* pp. 330-335., (2019, November).

[8] C.Mehra, A. K. Sharma, & A. Sharma, "Elucidating ransomware attacks in cyber-security.," *International Journal of Innovative Technology and Exploring Engineering,* vol. 9, no. 1, 2019.

[9] L. Alhathally and E. Alsuwat, "RESEARCH ARTICLE RANSOMWARE ATTACK DETECTION AND PREVENTION," *International Journal of Current Research,* vol. 12, no. 11, (November, 2020).

[10] A. Gangwar, "Ransomware Attacks: - Impact, Symptoms, Working, Preventive Measures and Response," *International Journal of Engineering and Advanced Technology (IJEAT),* vol. 9, no. 6, pp. 188-191, (August 2020).

Received 21/October/2023; Accepted 02/January/2024

DOI: https://doi.org/10.33103/uot.ijccce.24.2.3

[11] K. Shaukat, S. Luo, S. Chen, & D. Liu, "Cyber threat detection using machine learning techniques: A performance evaluation perspective.," *In 2020 international conference on cyber warfare and security (ICCWS)IEEE.*, pp. 1-6 , (2020, October).

[12] L. Amer, " Malware development threats with modern technologies.," *Cyber Security: A Peer-Reviewed Journal, ,* vol. 6, no. 2, pp. 178-187., 2022.

[13] R.Ali, A. Ali, F. Iqbal, M. Hussain, & F. Ullah, "Deep learning methods for malware and intrusion detection: A systematic literature review," *Security and Communication Networks, ,* 2022.

[14] S.Egunjobi, S. Parkinson, & A. Crampton, "Classifying ransomware using machine learning algorithms.," *In Intelligent Data Engineering and Automated Learning–IDEAL 2019: 20th International Conference, Manchester, UK, November 14–16, 2019, Proceedings, Part II 20. Springer International Publishing.,* pp. 45-52, 2019.

[15] K. Lee, S. Y. Lee, & K. Yim, "Machine learning based file entropy analysis for ransomware detection in backup systems," *IEEE Access* , vol. 7, pp. 110205-110215., 2019.

[16] K. Bakour, & H. M. Ünver, "VisDroid: Android malware classification based on local and global image features, bag of visual words and machine learning techniques.," *Neural Computing and Applications,* vol. 33, pp. 3133-3153, 2020.

[17] M. Basnet, S. Poudyal, M. H. Ali, & D. Dasgupta, "Ransomware detection using deep learning in the SCADA system of electric vehicle charging station," *n 2021 IEEE PES Innovative Smart Grid Technologies Conference-Latin America (ISGT Latin America)* , pp. 1-5, 2021, September.

[18] I. U. Haq, T. A. Khan, & A. Akhunzada, "A dynamic robust DL-based model for android malware detection," *IEEE Access,* vol. 9, pp. 74510-74521, 2021.

[19] M. S. Akhtar, & T. Feng, "Detection of malware by deep learning as CNN-LSTM machine learning techniques in real time," *Symmetry,* vol. 14, no. 11, p. 2308, 2022.

[20] Z. Namrud, S. Kpodjedo, A. Bali, , & C. Talhi, " Deep-layer clustering to identify permission usage patterns of android app categories.," *IEEE Access,* vol. 10, pp. 24240-24254., 2022.

[21] H. Madani, N. Ouerdi, A. Boumesaoud, & A. Azizi, "Classification of ransomware using different types of neural networks," *Scientific Reports,* vol. 12, no. 1, p. 4770, 2022.

[22] U. Zahoora, A. Khan, M. Rajarajan, , S. H. Khan, M. Asam & T. Jamal, "Ransomware detection using deep learning based unsupervised feature extraction and a cost sensitive Pareto Ensemble classifier.," *Scientific Reports,* vol. 12, no. 1, p. 15647, 2022.

[23] M. Masum, M. J. H. Faruk, H. Shahriar, K. D. I., "Ransomware classification and detection with machine learning algorithms.," *In 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)* , pp. 0316-0322, 2022, January.

[24] R. A. Mowri,M. Siddula and K. Roy, "Application of Explainable Machine Learning in Detecting and Classifying Ransomware Families Based on API Call Analysis," *arXiv:2210.11235v3 [cs.CR]* , 13 Nov 2022.

[25] N. Rani, & S. V. Dhavale, "Leveraging machine learning for ransomware detection.," *arXiv preprint arXiv:2206.01919.,* 2022.

[26] E. Berrueta, D. Morato, E. Magaña, & M. Izal, "Crypto-ransomware detection using machine learning models in file-sharing network scenarios with encrypted traffic," *Expert Systems with Applications, ,* vol. 209, p. 118299, 2022.

[27] J. Tang, R.Li, Y.Jiang, X. Gu, & Y.Li, "Android malware obfuscation variants detection method based on multi-granularity opcode features.," *Future Generation Computer Systems,* vol. 129, pp. 141-151, 2022.

[28] M. R. Amal, & P. Venkadesh, " Review of cyber attack detection: Honeypot system.," *Webology,* vol. 19, no. 1, pp. 5497-5514, 2022.

[29] K. Kim, F. A. Alfouzan, & H. Kim, "Cyber-Attack Scoring Model Based on the Offensive Cybersecurity Framework," *Applied Sciences,* vol. 11, no. 16, p. 7738, 2021.

[30] F. Haltaş, E. Uzun, N. Şişeci, A. Poşul, & B. Emre, "An automated bot detection system through honeypots for large-scale.," *In 2014 6th International Conference On Cyber Conflict (CyCon 2014),IEEE,* pp. 255-270, (2014, June).

[31] H. T. Nguyen, Q. D. Ngo, and V. H. Le, "IoT botnet detection approach based on PSI graph and DGCNN classifier," *in 2018 IEEE International Conference on Information Communication and Signal Processing (ICICSP)*, pp. 118-122, IEEE, 2018.

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

[33] O. P. Dwyer, A. K. Marnerides, V. Giotsas, & T. Mursch, "Profiling iot-based botnet traffic using dns," *In 2019 IEEE global communications conference (GLOBECOM),* pp. 1-6, (2019, December).

Received 21/October/2023; Accepted 02/January/2024

DOI: https://doi.org/10.33103/uot.ijccce.24.2.3

[34] R. Vishwakarma, & A. K. Jain, " A honeypot with machine learning based detection framework for defending IoT based botnet DDoS attacks," *In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI),IEEE,* pp. 1019-1024, (2019, April).

[35] S. Alhaidari, & M. Zohdy, "Hybrid learning approach of combining cluster-based partitioning and hidden markov model for iot intrusion detection," *In Proceedings of the 2019 3rd International Conference on Information System and Data Mining,* pp. 27-31, 2019.

[36] C. W. Tien, S. W. Chen, T. Ban, & S. Y. Kuo, "Machine learning framework to analyze iot malware using elf and opcode features.," *Digital Threats: Research and Practice,* vol. 1, no. 1, pp. 1-19, 2020.

[37] M. Shobana, & S. Poonkuzhali, "A novel approach to detect IoT malware by system calls using Deep learning techniques.," *In 2020 International Conference on Innovative Trends in Information Technology (ICITIIT),IEEE,* pp. 1-5, 2020.

[38] I. M. M. Matin, & B. Rahardjo, " The use of honeypot in machine learning based on malware detection: A review," *In 2020 8th International Conference on Cyber and IT Service Management (CITSM), IEEE,* pp. 1-6, (2020, October).

[39] X. Zhou, Y. Jin, H. Zhang, S. Li, & X. Huang, "A map of threats to validity of systematic literature reviews in software engineering.," *In 2016 23rd Asia-Pacific Software Engineering Conference (APSEC),* pp. 153-160, (2016, December).

[40] M. S. Yousaf, M. H. Durad, & M. Ismail, "Implementation of portable executable file analysis framework (PEFAF)," *In 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST),IEEE,* pp. 671-675, (2019, January).

[41] Veeraghattam, Kannappa, Mukkamala, & Sung., "Network Based Detection of Virtual Environments and Low Interaction Honeypots.," *In 2006 IEEE Information Assurance Workshop* , pp. 283-289, (2006, June).

[42] M. Nawrocki, M. Wählisch, T. C. Schmidt, C. Keil, & J. Schönfelder, "A survey on honeypot software and data analysis.," *arXiv preprint arXiv:1608.06249.,* 2016.

[43] C. Seifert, I. Welch, & P. Komisarczuk, "Taxonomy of honeypots.," *Victoria University of Wellington, Wellington.,* 2006.

[44] A. F. Assmaa, M. H. Hanaa, & F. H. . Int. J. Nidaa, "Design requirements of a smart phone honeypot system," *Comput. Appl,* vol. 6, no. 4, 2014.

[45] H. M. Ahmed, N. F. Hassan, & P. S. A. A. Fahad, "A survey on smartphone honeypot.," *International Journal of Computers & Technology,* vol. 11, no. 4, pp. 2476-2480, 2013.

[46] N. Provos, "Honeypot Background," HONEYD, [Online]. Available: https://www.honeyd.org/background/.

Received 21/October/2023; Accepted 02/January/2024