Unraveling Ransomware: Detecting Threats with Advanced Machine Learning Algorithms

Karam Hammadeh, M. Kavitha
Department of Computer Science and Engineering,
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India

***Abstract*—In our contemporary world, the pervasive influence of information technology, computer engineering, and the Internet has undeniably catalyzed innovation, fostering unparalleled economic growth and revolutionizing education. This technological juggernaut, however, has unwittingly ushered in a parallel era of new criminal frontiers, a magnet for hackers and cybercriminals. These malevolent actors exploit the vast expanse of electronic devices and interconnected networks to perpetrate an array of cybercrimes, and among these insidious digital threats, ransomware reigns supreme. Ransomware, characterized by its ominous ability to encrypt victims' data and extort payment for its release, stands as a dire menace to individuals and organizations alike. Operating with stealth and propagating with alarming alacrity through digital networks, ransomware has emerged as a formidable adversary in the digital age. This research paper focuses on the evolving stages of ransomware, driven by cutting-edge technologies, and proposes essential methods and ideas to detect and combat this menace. The proposed methodology, anchored in Cuckoo Sandbox, PE file feature extraction, and YARA rules, orchestrates three crucial phases: data collection, feature selection, and data preprocessing, all harmonizing to strengthen our defense against this concealed cyber menace. This paper contributes to the development of effective solutions for detecting and mitigating this hidden and insidious cyber threat. This work involves the application of multiple machine learning algorithms, including LSTM, which achieves an impressive accuracy of 99% in identifying ransomware attacks.**

Keywords—Ransomware; cuckoo sandbox; PEFile; YARA rules; machine learning; LSTM

I. Introduction

Cybersecurity has emerged as a crucial domain within information technology and computer engineering due to the rapid advancement and widespread adoption of technologies in our daily lives. As technology continues to evolve, it brings both positive and negative impacts, making it essential to protect data, preserve individual privacy, and safeguard innovation and intellectual property [1]. The primary objective of cybersecurity is to combat cybercrimes, which have significantly increased since the beginning of the 21st century.

Cybercrime is recognized as one of the most damaging and costly forms of criminal activity. Hackers and criminals exploit the power of networks, programming, and computers to steal valuable data, gain unauthorized access to bank accounts, and mass significant financial gains. Their illegal activities often remain hidden, making it challenging to trace the perpetrators and understand the extent of their actions. There are numerous methods through which individuals can become involved in cybercrime [2]. One such approach is the development and dissemination of malicious code, such as malware and ransomware, which can wreak havoc on computer systems and networks. Another technique employed by cybercriminals is launching Distributed Denial-of-Service (DDoS) attacks, which dominate servers and disrupt their ability to provide services.

Malware, or malicious software, is a wide-open problem that is difficult to solve in computer science [3]. It is a term used to describe various forms of harmful software designed to compromise computer systems, steal data, or disrupt normal operations. There are different types of malwares, including viruses, worms, Trojan horses, and ransomware [4,5]. Malware aims to change the behavior of either the operating system kernel or some security-sensitive applications without the user's consent, and all services of the system will therefore be undocumented. Some countries are developing this kind of malicious application in central intelligence for spying purposes; individuals or teams are also developing it for hijacking or showing their talent [6, 7].

Ransomware is a serious threat that poses a risk to individuals and organizations. It develops rapidly because it uses newer techniques like RSA and C&C servers. These techniques are difficult to be analyzed and to be detected (binary files, payload) [8, 9]. The first ransomware attack occurred in 1989. It was a trojan called PC Cyborg/AIDS, which was created to hide the folders and encrypt the names of all the files. It targeted the files associated with the ADIS conference, and restoration is achievable provided the filename and extension encryption tables are discovered [10].

At the beginning of this decade, ransomware used new techniques. In 2013, RSA 2048-bit was the main characteristic that was implemented with the public key algorithm for ransomware. After that, it could be used as a command and control (C&C) server to communicate through the Tor network, and it became able to target only specific types of file extensions [9, 11].

Encryption methods in ransomware have been significantly developed. For example, in 2015, most ransomware families were using the default configuration of AES file encryption and getting payment via Bitcoin. Later in 2017, the encryption became hybrid, using the AES algorithm to encrypt the files and RSA to encrypt the AES key [11]. According to [8, 12, 13], ransomware has been separated by researchers into many major types according to how it works methodologically and what kinds of effects that will cause. First, Crypto-Ransomware, Once the ransomware infects the target's device, the virus stealthily detects and encrypts the victim's important files, and the user will not be able to access, use, or get data until he pays the ransom. Second, Locker-Ransomware, which doesn't encrypt the data, has turned its focus to blocking access to the user’s equipment and information by disabling the user interface. Third, MBR-Ransomware works to encrypt the Master Boot Record table on the victim's PC, which means the files will be safe and not affected by encryption, but the system will not recognize the location of files and their mac-Ransomware and Mobile-Ransomware.

Some of the researchers [8, 14, 15] also found that there are three types of ransomware according to the style of encryption algorithm that has been used (see Fig. 1). These types are:

- Symmetric Ransomware: It uses symmetrical encryption algorithms like AES and DES to encrypt the victim’s data; in this type, the encryption and decryption use the same key.
- Asymmetric Ransomware: It uses an asymmetrical encryption algorithm, and ransomware is embedded with a public key to encrypt the victim's data, or it downloads during connection with a command and control (C&C) server, but the private key is saved only with the attacker, which is impossible to get without paying.
- Hybrid Ransomware: It uses symmetrical encryption to encrypt the victim's files, but the symmetric key will be encrypted by an asymmetric encryption algorithm. This technique takes advantage of both symmetric and asymmetric encryption.

The 2022 update of the Verizon Data Breach Investigations Report (DBIR), as shown in Fig. 2, reveals a concerning trend in the rise of ransomware attacks. According to the report, the number of ransomware incidents surged by 13% between 2020 and 2021, surpassing the combined increase of the previous five years. This significant escalation in ransomware incidents highlights the growing threat posed by cybercriminals targeting organizations across various sectors [16].

Ransomware detection relies on two core analyses: static and dynamic. Static analysis examines code without execution, offering security but struggling with packed or obfuscated malware. Dynamic analysis executes code in a controlled environment, effectively capturing behavioural patterns but posing security risks. Often, a hybrid approach combines these methods with others to maximize accuracy and minimize associated risks.

This research paper is driven by three core objectives. Firstly, it aims to establish a robust system for the early identification of ransomware threats, prioritizing swift detection to minimize potential damage inflicted upon data and computer systems. Secondly, the paper endeavors to raise awareness among individuals and organizations regarding the substantial risks and dire consequences associated with ransomware attacks, fostering a proactive and vigilant cybersecurity mindset. Lastly, a central focus of this research is the development and implementation of effective countermeasures and response strategies to mitigate the evolving menace of ransomware. By achieving these objectives, this work not only enhances our ability to combat ransomware but also contributes significantly to the broader realm of cybersecurity, fortifying our digital defences against this persistent and pernicious cyber threat.

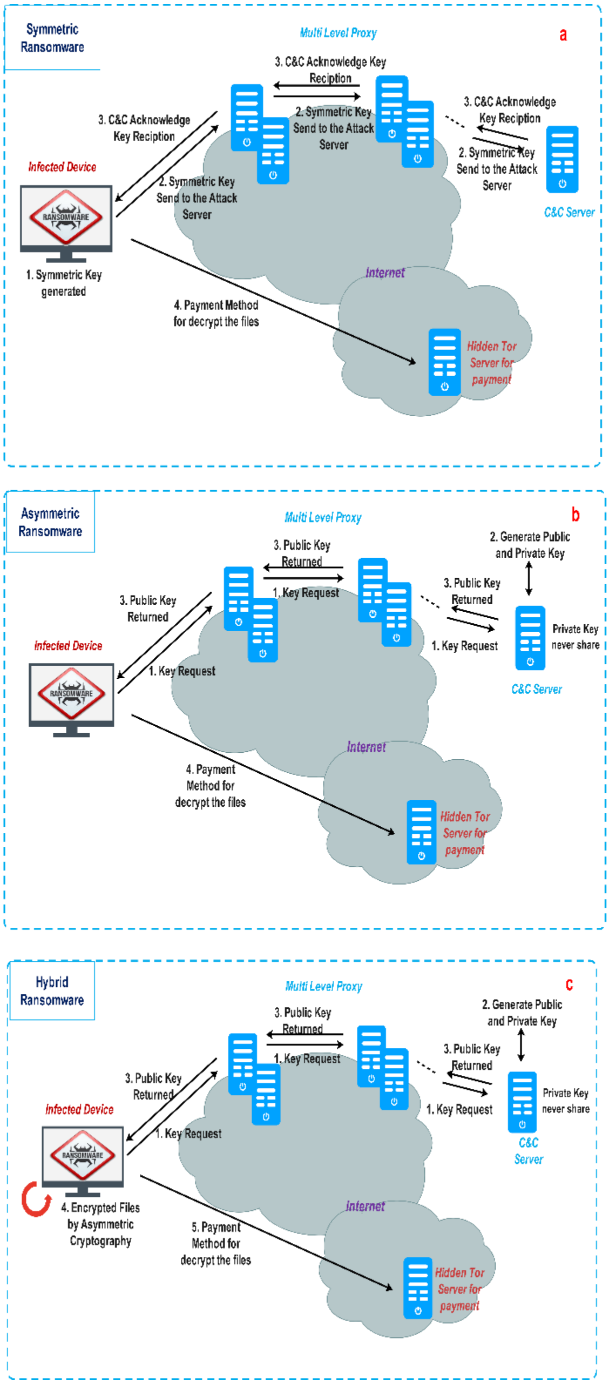

Symmetric Ransomware
Multi Level Proxy
a
3. C&C Acknowledge Key Reciption
3. C&C Acknowledge Key Reciption
2. Symmetric Key Send to the Attack Server
3. C&C Acknowledge Key Reciption
2. Symmetric Key Send to the Attack Server
Infected Device
2. Symmetric Key Send to the Attack Server
C&C Server
RANSOMWARE
1. Symmetric Key generated
Internet
4. Payment Method for decrypt the files
Hidden Tor Server for payment
Asymmetric Ransomware
Multi Level Proxy
b
3. Public Key Returned
1. Key Request
2. Generate Public and Private Key
3. Public Key Returned
3. Public Key Returned
Infected Device
1. Key Request
Private Key never share
1. Key Request
C&C Server
Internet
4. Payment Method for decrypt the files
Hidden Tor Server for payment
Hybrid Ransomware
Multi Level Proxy
c
3. Public Key Returned
1. Key Request
2. Generate Public and Private Key
3. Public Key Returned
3. Public Key Returned
Infected Device
1. Key Request
1. Key Request
Private Key never share
C&C Server
4. Encrypted Files by Asymmetric Cryptography
Internet
5. Payment Method for decrypt the files
Hidden Tor Server for payment

Fig. 1. Types of Ransomwares (a: Symmetric, b: Asymmetric, c: Hybrid).

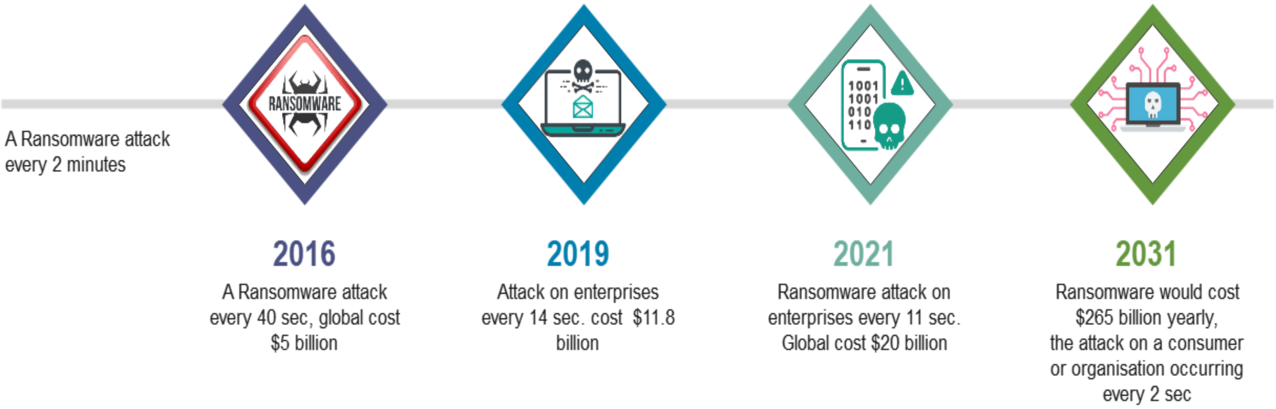

A Ransomware attack every 2 minutes
RANSOMWARE
1001 1001 010 110
2016
A Ransomware attack every 40 sec, global cost $5 billion
2019
Attack on enterprises every 14 sec. cost $11.8 billion
2021
Ransomware attack on enterprises every 11 sec. Global cost $20 billion
2031
Ransomware would cost $265 billion yearly, the attack on a consumer or organisation occurring every 2 sec

Fig. 2. DBIR and cybersecurity ventures report.

II. Related Work

Palisse et al. [17] recommended using Cryptographic approaches for proactively protecting against Ransomware. where ransomware constructs a dynamic block for calls to cryptographic APIs based on weaknesses and exploits vulnerabilities in cipher modes of operation. A mechanism termed PAYBREAK was suggested by Kolodenkerz et al. [18]. Because the system logs all of the random numbers it generates in a massive log file or database, users may use this information to exhaustively search for encryption keys. That method has proven to be quite effective. Kim et al. [19] used the same methods as the earlier researchers and developed a Deterministic Random Bit Generator (DRBG) to thwart ransomware; the DRBG is used to generate a seed, which is then combined with user DRBG data to produce a decryption key.

Poudyal et al. [20] suggested using reverse engineering to pull elements like assembly instructions and DLLs, which can subsequently be used with machine learning methods like K-fold Cross-Validation to identify ransomware. Ahmadian et al. [14] employed a system that relies on the Connection Monitor & Connection Breaker (CMCB) to identify and detect stealthy ransomware by monitoring and analysing network activity. Tseng et al. [21] and Cabaj et al. [22] conducted researches on the analysis and detection of ransomware attacks from different perspectives. They focused on analysing the HTTP and TCP protocols to identify ransomware attacks. Tseng utilised deep learning techniques for this purpose. On the other hand, Cabaj designed an approach based on Software-Defined Networking (SDN) to both detect and mitigate ransomware attacks. Cusack et al. [23] and Al Mashhadani et al. [8] proposed a framework for monitoring and analyzing data traffic during a ransomware attack. Their approach involved intercepting the communication between an infected machine, unaffected devices, and the command-and-control (C&C) server using a network protocol analyzer like Wireshark. To ensure the security of the monitoring system, they recommended implementing a firewall to shield it from potential threats. Shakir et al. [24] delved into the evolution of ransomware, tracing its development from phone antivirus and deceptive software to the emergence of crypto-ransomware. The study revealed two crucial factors driving the increase in ransomware attacks. Firstly, tracking victim payments to attackers proved to be challenging due to the use of anonymous channels, making it challenging to trace and apprehend the perpetrators. Secondly, the practicality and effectiveness of employing cryptographic technologies played a significant role in the surge of ransomware attacks. These findings shed light on the key catalysts behind the proliferation and sophistication of ransomware as a malicious threat. The approach of Homayoun et al. [25] relied on the monitoring of activity records to detect and identify ransomware attacks by utilising machine learning methods to identify specific patterns. By analysing and examining activity logs, which capture system and user behaviour, it becomes possible to uncover anomalous patterns that indicate the presence of ransomware whereas Medhat et al. [26] developed a novel framework that relies on a static analysis approach. The framework utilises YARA rules, which are a set of predefined rules for pattern matching, to extract feature rules for each file. By applying these rules and conducting a classification process, the framework can identify files or processes that exhibit ransomware-like behaviour. The details and specific workings of this framework are represented in Fig. 3.

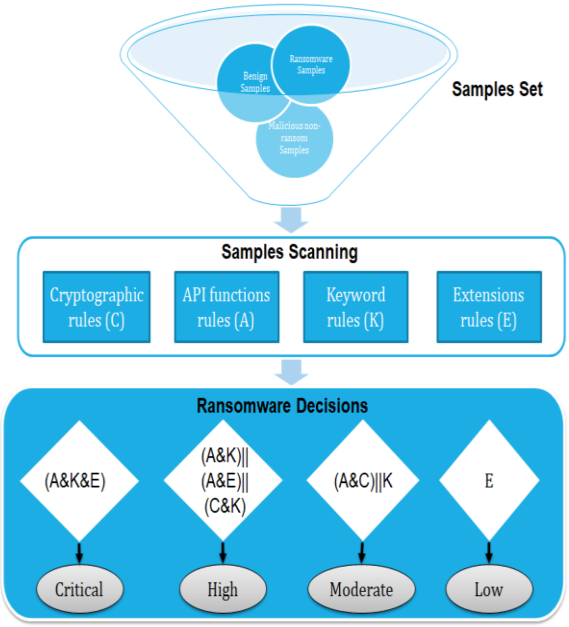


Fig. 3. Static-based framework for ransomware detection [25].

Detecting ransomware is a complex task due to its ever-changing nature, requiring researchers and security professionals to employ diverse strategies. One recommended approach involves monitoring the Master File Table (MFT) for any unusual activity or modifications, as ransomware often targets and encrypts files. Another approach involves leveraging machine learning models to search for specific language patterns associated with ransomware. By analyzing text data using these models, suspicious indicators can be identified. Additionally, examining data flow and network traffic can help detect ransomware [27]. Moore et al. [28] introduced a novel method for detecting ransomware utilising honeypot technology, as depicted in Fig. 4. This approach encompasses several stages to enhance detection capabilities. In the behaviour stage, two tools, namely DatAdvantage and HitmanPro, are utilised. DatAdvantage employs user behaviour analytics to identify abnormal activities, while HitmanPro focuses on detecting unusual system behaviour. The network stage involves monitoring data traffic across the network to detect any exchange of file keys, a common characteristic of ransomware operations. Lastly, in the server stage, changes are monitored using the file server resource manager, and a file screening function is employed to control access and block the writing of unauthorised files. This multi-stage approach aims to provide a comprehensive detection mechanism, combining behavioural analysis, network monitoring, and server-level control to enhance ransomware detection and prevention capabilities.

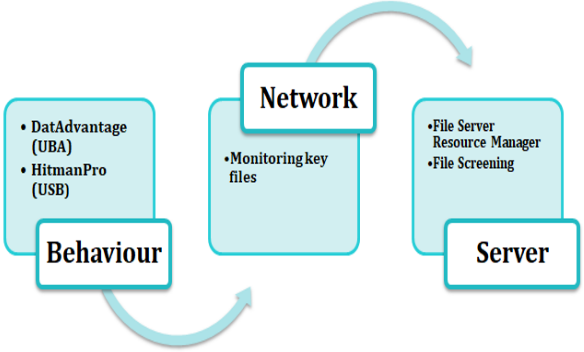


Fig. 4. Honeypot techniques to detect ransomware [28].

Ransomware can be classified into two main types: non-spreading ransomware and ransomware with worm-like characteristics. The primary objective in combating ransomware is to prevent the victim's device from being encrypted and to halt the malware's spread within and beyond the local network. Cabaj et al. [29] utilised SDN to detect non-spreading ransomware variants like CryptoWall and Locky. Their approach involved using the size of the first three HTTP Post packets in sequence as a detection mechanism. Through their implementation, they achieved a high true positive rate of 97-98%. In the case of ransomware that spreads like worms, such as WannaCry and ExPetr, researchers in [30] have suggested utilising two programmes for effective mitigation. The first programme detects and prevents WannaCry from encrypting the victim's device by monitoring and blocking communications between the ransomware and its Command-and-Control (C&C) servers through a dynamic IP blacklist. The second programme tracks the ports used by WannaCry, enabling the prevention of encryption processes and the identification and containment of the ransomware's spreading behaviour.

III. METHODOLOGY

The methodology (see Fig. 5) describes a technique for detecting ransomware using Cuckoo Sandbox [31], feature extraction from Portable Executable (PE) files [32], and YARA rules [33]. By analysing the characteristics and patterns of PE files, it is possible to identify potential ransomware threats and lessen their impact. There are three main stages to prepare data to be consumed by the algorithms we test: data collection, feature selection, and data processing. Data Collection is contingent upon collecting a diverse set of PE files,This investigation encompasses both benign components, which include application and system files that form the foundation of our digital activities, as well as malicious ransomware samples obtained from VirusShare, and utilising YARA rules to detect crypto signatures. Feature Selection is applied to analyse extracted characteristics and choose the most useful ones for ransomware detection. We employ Weight of Evidence (WoE) and Information Value (IV). They are used to determine a dataset's predictive potential and select which elements are most significant for a modelling task. Utilising data preprocessing techniques, such as normalisation, dimensionality reduction, data reshaping, and feature scaling is to improve the performance of subsequent machine learning algorithms.

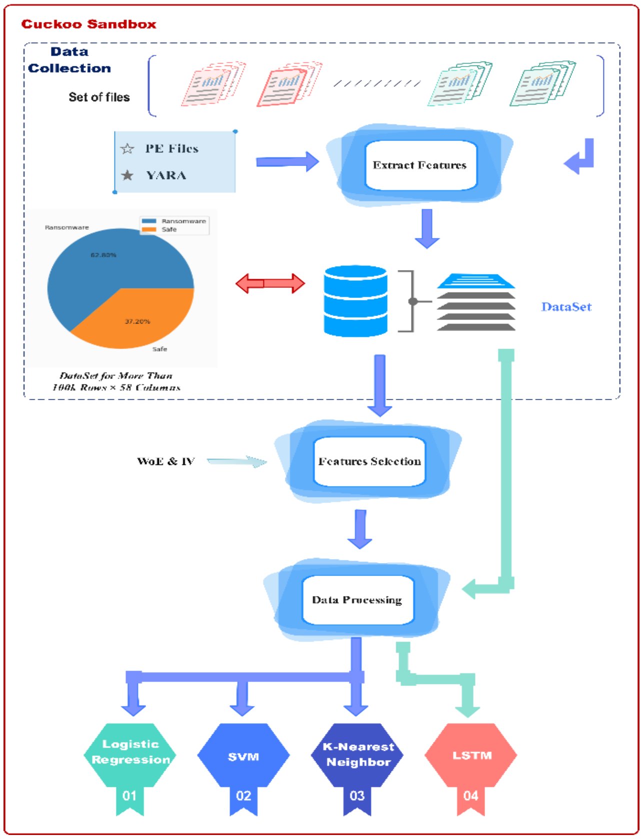


Fig. 5. Proposed methodology for detecting ransomware.

The output of the data processing stage is now suitable to be injected into the algorithms we test. We apply four machine learning algorithms: Logistic Regression, SVM, K-Nearest Neighbor, and LSTM.

Logistic Regression [34] is a type of generalized linear model that is used for classification tasks. The goal of logistic regression is to find the best model that describes the relationship between a set of input features and a binary output variable. The model is represented by a logistic function, which maps the input features to a value between 0 and 1, representing the probability that the output variable is in one of the two classes. In logistic regression, a linear combination of input features is transformed by the logistic function [35], which is also known as the sigmoid function. The coefficients of the linear combination are learned from the training data using a technique called maximum likelihood estimation. Once the model is trained, new data can be input into the model, and the output of the logistic function can be used to predict the probability that the new data belongs to one of the two classes. It is also less prone to overfitting compared to more complex models [36].

Support vector machines (SVMs) can solve classification and regression issues [37]. SVMs find the best boundary for categorising data. This border was chosen to maximise the margin, which is the distance between the boundary and each class's closest data points. After identifying this barrier, additional data may be categorised by its side [38]. SVMs are useful when data is not linearly separable, meaning classes cannot be separated by a straight line. SVMs use the "kernel trick" to shift data into a higher-dimensional space where it may be linearly segregated [39].

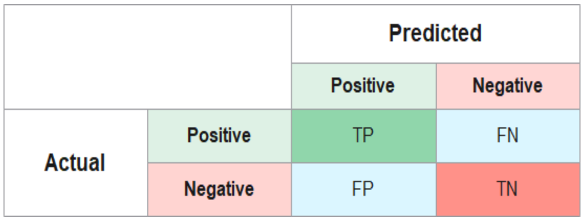
The K-Nearest Neighbors (K-NN) is a kind of supervised learning that classifies unknown files in line with previously established categories [40]. Among the many categorization algorithms, K-NN is a useful algorithm for grouping unknown objects into categories with the greatest number of shared attributes [41]. K-NN presupposes that close data points belong to the same class, which may not always be true. The data distribution, characteristics, and distance metric affect K-NN's efficacy [42].

Long-short term memory (LSTM), a standard and enhanced recurrent neural network, is capable of analysing and anticipating time series problems. A memory cell composed of an input gate, a forget gate, and an output gate regulates the transmission of information in the LSTM model. The problem of expanding gradients and vanishing is resolved by LSTM's unique structure [43].

IV. Evaluation and Results

In order to evaluate the performance of classification models, various evaluation measures are commonly utilized. These measures include accuracy, precision, recall, and the F1 Score. The confusion matrix, represented as Fig. 6, is used to assess the classifier's primary performance indicators.

Accuracy, as expressed by Eq. (1) represents the percentage of correctly classified instances in relation to the entire dataset. Precision, or Detection Rate, as given by Eq. (2), is commonly used when dealing with imbalanced datasets. It measures the proportion of correctly classified instances compared to the total instances that are correctly and incorrectly classified. Recall, described in Eq. (3), is the percentage of accurately classified instances in relation to the total number of actual positive instances. The False Alarm Rate, mentioned in Eq. (4), indicates the frequency with which attacks are misclassified or falsely identified. Finally, the F1 Score or F-measure, described by Eq. (5), provides an overall assessment of the accuracy of a test by combining precision and recall into a single metric. This provides insights into the classifier's primary performance indicators and facilitates further analysis and comparison of different models and techniques.



		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Fig. 6. Confusion matrix.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$False\_Alarm\_Rate = \frac{FP}{FP+TN} \quad (4)$$

$$F1\_Score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (5)$$

In feature selection, we have used WOE and IV as criteria for ranking and selecting variables. Variables with high IV values are considered more predictive and are more likely to be included in the final set of features, as shown in Fig. 7. By focusing on variables with strong predictive power, it can reduce dimensionality and improve the performance and interpretability of the used models.

Table I and Fig. 8 present a comprehensive overview of the performance results of different models utilized for ransomware detection, including LSTM, SVM, LR, and KNN. These findings offer valuable insights into the capabilities and effectiveness of each model in addressing the task at hand. Since the dataset consists of sequences of bits, LSTM's ability to remember and learn from previous information makes it a valuable choice. By utilizing its memory cells and gating mechanisms, the LSTM model can effectively process and interpret the sequential nature of the data, making it well-suited for the task at hand.

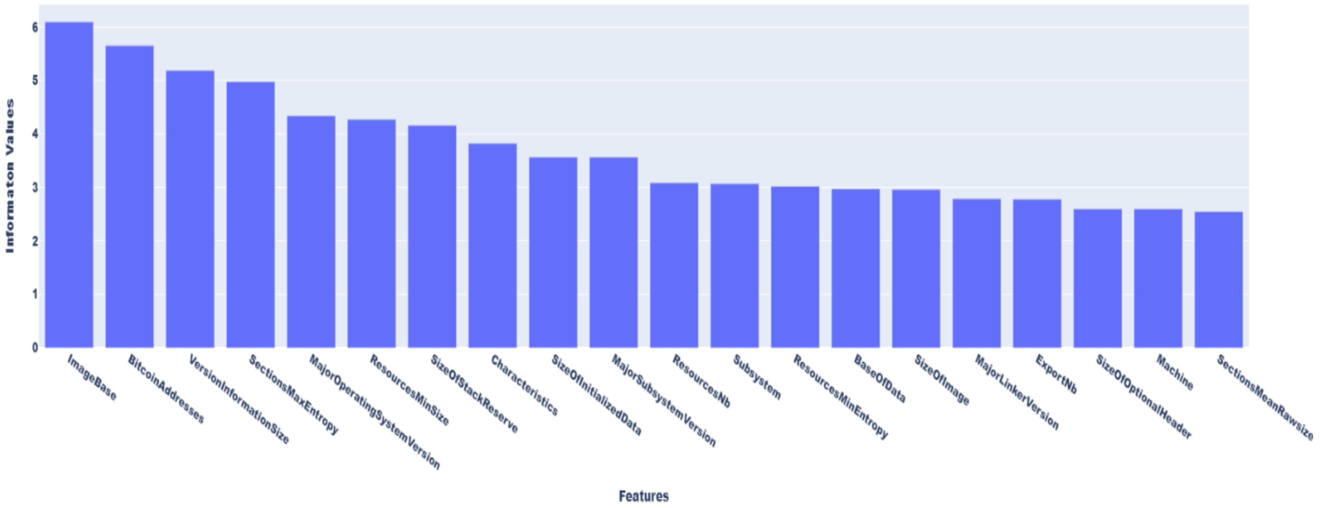


Fig. 7. Importanc of features based on information value.

TABLE I. EXPERIMENTAL RESULTS

	LSTM	SVM	LR	KNN
TN	28689	28651	28342	28380
FN	235	283	592	494
FP	146	378	640	281
TP	12345	12103	11841	12260
Totall	41415	41415	41415	41415
Accuracy	**0.9908**	**0.98404**	**0.970252**	**0.981287**

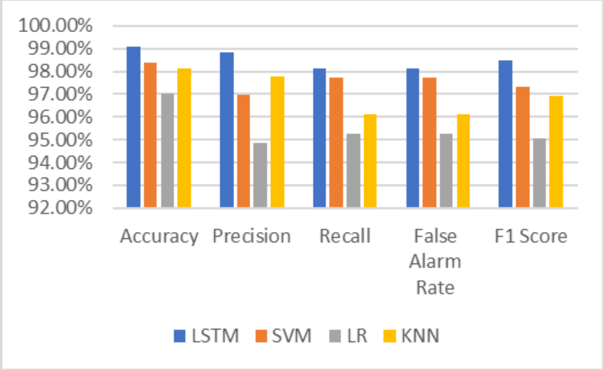


Fig. 8. Comprised the results of applied algorithms.

Support Vector Machine (SVM) and logistic regression face challenges when dealing with massive datasets due to their computational complexity, leading to longer training times as the dataset size increases. However, for smaller datasets and lower-dimensional feature spaces, K-Nearest Neighbors (K-NN) remains a valuable and effective classification technique, boasting an impressive accuracy of 98%. K-NN's simplicity and ability to capture patterns make it a practical choice for such scenarios. The LSTM model exhibited the highest accuracy, achieving an impressive score of 0.9908. This indicates that the LSTM model is highly effective in detecting ransomware attacks, showcasing its ability to capture intricate patterns and temporal dependencies within the data. With such a high accuracy, the LSTM model can be considered a robust choice for ransomware detection. During the training process, the LSTM model continuously updates its internal parameters to minimize the loss, resulting in better predictions and higher accuracy over time. As the model converges, the accuracy typically improves, and the loss decreases as shown in Fig. 9.

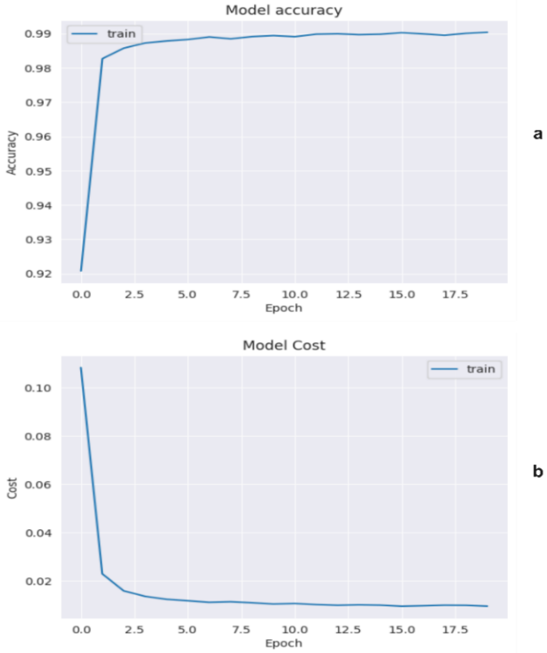


Fig. 9. Visualized results of LSTM (a. Accuracy & b. Loss mode).

V. CONCLUSION AND FUTURE WORKS

In conclusion, the studies on ransomware detection have generally yielded positive results. Researchers have identified distinct static features in ransomware samples, such as cryptographic signatures, API methods, and file extensions,that can aid in their identification. Different approaches, including the use of surveillance and honeypots, have been explored to track down and analyze ransomware. Machine learning and classification techniques have proven to be valuable in enhancing both static and dynamic malware analysis.

Additionally, the application of deep learning, such as the LSTM model, has shown remarkable accuracy, reaching up to 99%, in detecting malware and ransomware. These findings highlight the potential of advanced techniques for improving cybersecurity measures against ransomware threats. Continued research and development in this field can further strengthen the detection and mitigation of ransomware attacks. In future work, we propose extending the dataset used for ransomware detection to include a more comprehensive set of features derived from both dynamic and static analysis. Moreover, we intend to explore the use of a hybrid algorithm combining CNN-LSTM models. This fusion of techniques has the potential to improve the accuracy and robustness of ransomware detection, paving the way for more effective defense mechanisms against evolving ransomware threats.

REFERENCES

[1] M. Robinson, K. Jones, and H. Janicke, “Cyber warfare: Issues and challenges,” Computers & Security, vol. 49, pp. 70–94, Mar. 2015, doi: https://doi.org/10.1016/j.cose.2014.11.007.

[2] T. J. Holt and A. M. Bossler, Cybercrime in progress : theory and prevention of technology-enabled offenses. London ; New York: Routledge, 2016.

[3] E. Filiol, M. Helenius, and S. Zanero, “Open Problems in Computer Virology,” Journal in Computer Virology, vol. 1, no. 3–4, pp. 55–66, Feb. 2006, doi: https://doi.org/10.1007/s11416-005-0008-3

[4] M. Sikorski and A. Honig, Practical malware analysis : the hands-on guide to dissecting malicious software. San Francisco No Starch Press, 2012.

[5] T. Mane, Prachi Nimase, Prahalad Parihar, and Pragati Chandankhede, “Review of Malware Detection Using Deep Learning,” pp. 255–262, Oct. 2021, doi: https://doi.org/10.1007/978-981-16-5301-8_19.

[6] J. Rutkowska, “Introducing Stealth Malware Taxonomy,” 2006.

[7] A. Razgallah, R. Khoury, S. Hallé, and K. Khanmohammadi, “A survey of malware detection in Android apps: Recommendations and perspectives for future research,” Computer Science Review, vol. 39, p. 100358, Feb. 2021, doi: https://doi.org/10.1016/j.cosrev.2020.100358.

[8] A. O. Almashhadani, M. Kaiiali, S. Sezer, and P. O’Kane, “A Multi-Classifier Network-Based Crypto Ransomware Detection System: A Case Study of Locky Ransomware,” IEEE Access, vol. 7, pp. 47053–47067, 2019, doi: https://doi.org/10.1109/access.2019.2907485.

[9] Monika, P. Zavarsky, and D. Lindskog, “Experimental Analysis of Ransomware on Windows and Android Platforms: Evolution and Characterization,” Procedia Computer Science, vol. 94, pp. 465–472, 2016, doi: https://doi.org/10.1016/j.procs.2016.08.072.

[10] K. Lee, K. Yim, and J. T. Seo, “Ransomware prevention technique using key backup,” Concurrency and Computation: Practice and Experience, vol. 30, no. 3, p. e4337, Oct. 2017, doi: https://doi.org/10.1002/cpe.4337.

[11] P. O’Kane, S. Sezer, and D. Carlin, “Evolution of ransomware,” IET Networks, vol. 7, no. 5, pp. 321–327, 2018.

[12] H. Orman, “Evil Offspring - Ransomware and Crypto Technology,” IEEE Internet Computing, vol. 20, no. 5, pp. 89–94, Sep. 2016, doi: https://doi.org/10.1109/mic.2016.90.

[13] N. Aldaraani and Z. Begum, “Understanding the impact of Ransomware: A Survey on its Evolution, Mitigation and Prevention Techniques,” IEEE Xplore, pp. 1–5, Apr. 2018, doi: https://doi.org/10.1109/NCG.2018.8593029.

[14] M. M. Ahmadian, H. R. Shahriari, and S. M. Ghaffarian, “Connection-monitor & connection-breaker: A novel approach for prevention and detection of high survivable ransomwares,” 2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC), Sep. 2015, doi: https://doi.org/10.1109/iscisc.2015.7387902.

[15] A. Liska and T. Gallo, Ransomware : defending against digital extortion. Sebastopol (Calif.): O’reilly Media. Copyright, 2016.

[16] A. Fagioli, “Zero-day recovery: the key to mitigating the ransomware threat,” Computer Fraud & Security, vol. 2019, no. 1, pp. 6–9, Jan. 2019, doi: https://doi.org/10.1016/s1361-3723(19)30006-5.

[17] A. Palisse, H. Le Bouder, J.-L. Lanet, C. Le Guernic, and A. Legay, “Ransomware and the Legacy Crypto API,” Lecture Notes in Computer Science, vol. 10158, pp. 11–28, 2017, doi: https://doi.org/10.1007/978-3-319-54876-0_2.

[18] E. Kolodenker, W. Koch, G. Stringhini, and M. Egele, “PayBreak : Defense Against Cryptographic Ransomware,” Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Apr. 2017, doi: https://doi.org/10.1145/3052973.3053035.

[19] H. Kim, D. Yoo, J. -S. Kang and Y. Yeom, "Dynamic ransomware protection using deterministic random bit generator," 2017 IEEE Conference on Application, Information and Network Security (AINS), Miri, Malaysia, 2017, pp. 64-68, doi: 10.1109/AINS.2017.8270426.

[20] S. Poudyal, K. P. Subedi, and D. Dasgupta, “A Framework for Analyzing Ransomware using Machine Learning,” IEEE Xplore, Nov. 01, 2018.

[21] A. Tseng, Y. Chen, Y. Kao, and T. Lin, “Deep Learning for Ransomware Detection,” IEICE Technical Report; IEICE Tech. Rep., vol. 116, no. 282, pp. 87–92, Oct. 2016,

[22] K. Cabaj and W. Mazurczyk, “Using Software-Defined Networking for Ransomware Mitigation: The Case of CryptoWall,” IEEE Network, vol. 30, no. 6, pp. 14–20, Nov. 2016, doi: https://doi.org/10.1109/mnet.2016.1600110nm.

[23] G. Cusack, O. Michel, and E. Keller, “Machine Learning-Based Detection of Ransomware Using SDN,” Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, pp. 1–6, Mar. 2018, doi: https://doi.org/10.1145/3180465.3180467.

[24] H. A. Shakir and A. N. Jaber, “A Short Review for Ransomware: Pros and Cons,” Advances on P2P, Parallel, Grid, Cloud and Internet Computing, pp. 401–411, Nov. 2017, doi: https://doi.org/10.1007/978-3-319-69835-9_38.

[25] S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, and R. Khayami, “Know Abnormal, Find Evil: Frequent Pattern Mining for Ransomware Threat Hunting and Intelligence,” IEEE Transactions on Emerging Topics in Computing, vol. 8, no. 2, pp. 341–351, Apr. 2020, doi: https://doi.org/10.1109/tetc.2017.2756908.

[26] M. Medhat, S. Gaber, and N. Abdelbaki, “A New Static-Based Framework for Ransomware Detection,” 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), Aug. 2018, doi: https://doi.org/10.1109/dasc/picom/datacom/cyberscitec.2018.00124.

[27] N. Andronio, S. Zanero, and F. Maggi, “HelDroid: Dissecting and Detecting Mobile Ransomware,” Research in Attacks, Intrusions, and Defenses, pp. 382–404, 2015, doi: https://doi.org/10.1007/978-3-319-26362-5_18.

[28] C. Moore, “Detecting Ransomware with Honeypot Techniques,” 2016 Cybersecurity and Cyberforensics Conference (CCC), Aug. 2016, doi: https://doi.org/10.1109/ccc.2016.14.

[29] K. Cabaj, M. Gregorczyk, and W. Mazurczyk, “Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics,” Computers & Electrical Engineering, vol. 66, pp. 353–368, Feb. 2018, doi: https://doi.org/10.1016/j.compeleceng.2017.10.012.

[30] M. Akbanov, V. G. Vassilakis, and M. D. Logothetis, “Ransomware detection and mitigation using software-defined networking: The case of WannaCry,” Computers & Electrical Engineering, vol. 76, pp. 111–121, Jun. 2019, doi: https://doi.org/10.1016/j.compeleceng.2019.03.012.

[31] L. Wang, B. Wang, J. Liu, Q. Miao, and J. Zhang, “Cuckoo-based Malware Dynamic Analysis,” International Journal of Performability Engineering, 2019, doi: https://doi.org/10.23940/ijpe.19.03.p6.772781.

[32] Y. Zhang, X. Chang, Y. Lin, J. Misic, and V. B. Misic, “Exploring Function Call Graph Vectorization and File Statistical Features in Malicious PE File Classification,” IEEE Access, vol. 8, pp. 44652–44660, 2020, doi: https://doi.org/10.1109/access.2020.2978335.

[33] N. Naik et al., "Embedded YARA rules: strengthening YARA rules utilising fuzzy hashing and fuzzy rules for malware analysis," Complex & Intelligent Systems, vol. 7, no. 2, pp. 687–702, Nov. 2020, doi: https://doi.org/10.1007/s40747-020-00233-5.

[34] J. M. Hilbe, Logistic Regression Models. Chapman and Hall/CRC, 2009. doi: https://doi.org/10.1201/9781420075779.

[35] Z. Akram, M. Majid, and S. Habib, "A Systematic Literature Review: Usage of Logistic Regression for Malware Detection," IEEE Xplore, Nov. 01, 2021. https://ieeexplore.ieee.org/document/9693035 (accessed Jan. 12, 2023).

[36] Ö. A. Aslan and R. Samet, "A Comprehensive Review on Malware Detection Approaches," IEEE Access, vol. 8, pp. 6249–6271, Jan. 2020, doi: https://doi.org/10.1109/ACCESS.2019.2963724.

[37] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," Neurocomputing, vol. 408, pp. 189–215, Sep. 2020, doi: https://doi.org/10.1016/j.neucom.2019.10.118.

[38] M. Wadkar, F. Di Troia, and M. Stamp, "Detecting malware evolution using support vector machines," Expert Systems with Applications, vol. 143, p. 113022, Apr. 2020, doi: https://doi.org/10.1016/j.eswa.2019.113022.

[39] P. O'Kane, S. Sezer, K. McLaughlin, and E. G. Im, "SVM Training Phase Reduction Using Dataset Feature Filtering for Malware Detection," IEEE Transactions on Information Forensics and Security, vol. 8, no. 3, pp. 500–509, Mar. 2013, doi: https://doi.org/10.1109/tifs.2013.2242890.

[40] F. A. Narudin, A. Feizollah, N. B. Anuar, and A. Gani, "Evaluation of machine learning classifiers for mobile malware detection," Soft Computing, vol. 20, no. 1, pp. 343–357, Nov. 2014, doi: https://doi.org/10.1007/s00500-014-1511-6.

[41] D. Stiawan, S. M. Daely, A. Heryanto, N. Afifah, M. Y. Idris, and R. Budiarto, "Ransomware Detection Based On Opcode Behavior Using K-Nearest Neighbors Algorithm," Information Technology and Control, vol. 50, no. 3, pp. 495–506, Sep. 2021, doi: https://doi.org/10.5755/j01.itc.50.3.25816.

[42] H. A. Abu Alfeilat et al., "Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review," Big Data, vol. 7, no. 4, pp. 221–248, Dec. 2019, doi: https://doi.org/10.1089/big.2018.0175.

[43] R. Lu, "Malware Detection with LSTM using Opcode Language," arXiv:1906.04593 [cs], Jun. 2019