ISSN 2614-0047

Ransomware detection: patterns, algorithms, and defense strategies

Manar Y. Amro [a,1], Mohamed Dwieb [a,2], Jehad A.H Hammad [a,3,*] , Aji Prasetya Wibawa [b,4]

[a] Faculty of Technology and applied Sciences, Al-Quds Open University (QOU), Palestine
[b] Department of Electrical Engineering, Universitas Negeri Malang, Malang, Indonesia
[1] Manar_amro@hotmail.com; [2] mdweib@qou.edu; [3] jhammad@qou.edu; [4] aji.prasetya.ft@um.ac.id
* corresponding author

ARTICLE INFO

Article history
Received January 12, 2024
Revised February 15, 2024
Accepted May 8, 2024

Keywords
Ransomware detection
random forest
Cyber security
Neural network
Logistic regression
LSTM
Defense strategies

ABSTRACT

In the contemporary digital landscape, rapid technological advancements present unprecedented challenges for developers in the hardware and software realms. The ubiquitous presence of the Internet, the Internet of Things (IoT), and widespread digital solutions bring numerous benefits and escalating risks. This study investigates the pervasive threat of ransomware attacks, a daily menace that imperils the operational and security dimensions of the digital sphere for enterprises and individuals. The research objective is to identify the most effective algorithm for detecting ransomware viruses, a persistent and evolving threat that significantly challenges institutions, companies, and governmental organizations. The dynamic nature of ransomware necessitates robust detection mechanisms to safeguard sensitive data. To achieve this goal, we conducted a comparative analysis of four prominent algorithms recognized for their efficacy in combating and detecting viruses. Emphasis was placed on the algorithm exhibiting the most promising results. A detailed examination of its impact on existing data involved comprehensive analysis and a comparative assessment against previous studies. Results, derived from extensive studies and experiments on a diverse dataset, illuminate the critical role of ransomware detection algorithms and underscore their effectiveness. The findings contribute valuable insights to the ongoing discourse on cybersecurity strategies, providing a foundation for enhanced ransomware defense measures.

This is an open access article under the CC–BY-SA license.

1. Introduction

The ransomware lifecycle comprises seven key stages, demonstrating the formation of a cybercriminal ecosystem characterized by a collaborative relationship between the 'creator' and 'campaigner' [1]. The creator, responsible for developing ransomware code, cooperates closely with the campaigner, who orchestrates the attacking campaign [2]. This collaboration facilitates continuous improvement in knowledge and skills with each cycle, ultimately producing specialized criminals.

The stages include creation, involving the development and enhancement of ransomware codes; campaign, focused on disseminating the ransomware to individual and institutional victims through various infection vectors; infection, where the ransomware setup behavior begins; command and control, involving communication with a central server for encryption keys and additional files; search, targeting valuable files; encryption, utilizing various encryption technologies; and extortion, culminating in the display of a ransom demand specifying payment details and consequences for non-compliance [3]–[5]. The lifecycle underscores the strategic and coordinated efforts employed by cybercriminals throughout the ransomware attack process [6]. Both academic researchers and industrial security experts have proposed various Ransomware detection techniques. Several of these methods are currently in use. They primarily involve static or dynamic analysis of executables

suspected to be ransomware. Static analysis involves examining the code without actually running the executable, encompassing tasks such as static linking, identifying ASCII strings, packer detection, and analyzing memory relocation [7]. On the other hand, dynamic analysis takes place after the suspected ransomware is executed. This process records the actions and system calls made by the executable during its execution, forming the basis for generating a comprehensive report [8].

In this study, we will use four algorithms for the detection of ransomware, which is often applied in dynamic analysis scenarios. The model can learn from the behavior of files during execution, allowing it to identify patterns associated with ransomware activities. This approach enhances the ability to detect novel and evolving ransomware strains that may not be easily captured through static analysis alone.

2. Literature Review

Ransomware-as-a-Service (RaaS) was accessible through the Dark Web [9]. Successful ransomware attacks present a significant cybersecurity challenge, particularly in the era of widespread connectivity via the Internet of Things (IoT) [10]. The risks and impacts are notably elevated, especially for medical IoT devices [11].

The propagation of ransomware is largely attributed to the absence of cyber hygiene practices at the individual level [12]. Cyber hygiene encompasses various facets of online safety [13], such as responsible browsing habits, regular updating of antivirus software, cautious installation of third-party software, and maintaining user awareness. Adherence to cyber hygiene practices is crucial to preventing ransomware and other malware. Despite advancements in security standards and protocols, ransomware families have effectively infiltrated the defense systems of organizations, governments, and individual users, with common sources including email attachments, Removable Media, Malvertising, social media and SMS, and Ransomware as a Service) [8]. In general, ransomware exists in several forms, and Fig. 1 outlines a timeline featuring key families. For instance, locker ransomware is specifically designed to lock users out of their devices, coercing them into making payments. Conversely, cryptographic ransomware takes a different approach by encrypting user files and demanding a ransom, making it the most prevalent type in this category.

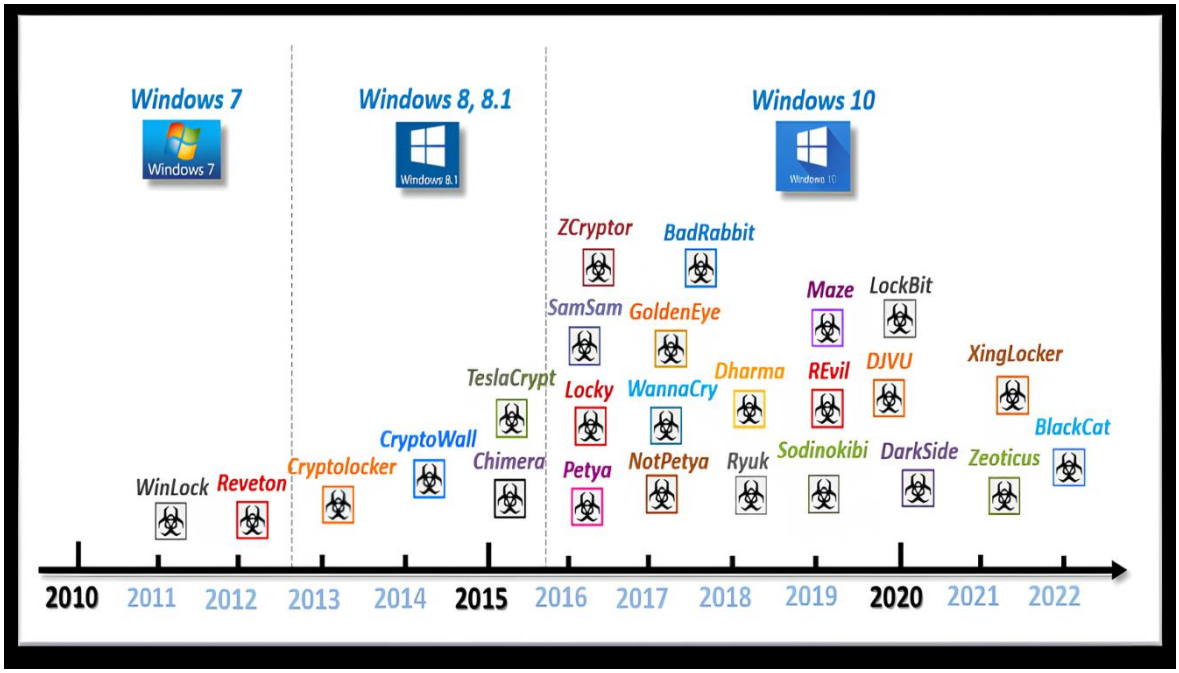

Fig. 1. Timeline of ransomware families (Windows-based)

Finaly, double-extortion ransomware adds the threat of data release, also known as doxing. The ransomware operation follows a sequence of stages referred to as the "kill-chain," illustrated in Fig. 2. While different versions of this sequence exist with varying stages and names presented by others,

the fundamental operations remain consistent, as briefly detailed [14]. Preventing ransomware presents notable challenges for several reasons.

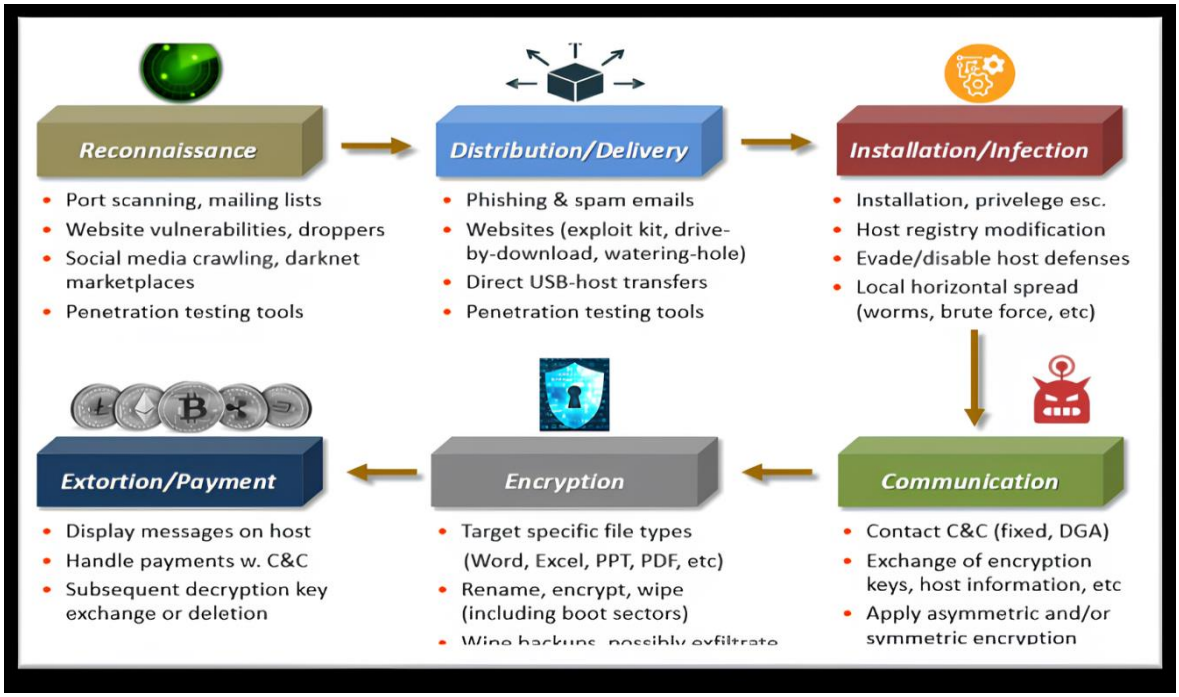

Fig. 2. Overview of ransomware attack "kill chain"

Ransomware operates similarly to benign software, operating discreetly [15]. Detecting ransomware in zero-day attacks is crucial. Primary objectives include averting system damage caused by ransomware, identifying zero-day malware, and minimizing detection errors—reducing false positives while still capturing all instances of ransomware. False positives occur when the system mistakenly identifies a harmless program or file as ransomware, triggering unnecessary alerts and actions. Various tools and methodologies are employed for ransomware detection. Static analysis methods, decomposing source code without execution, generate numerous false positives and struggle with disguised ransomware. Attackers frequently create new variations and modify codes using diverse packaging techniques. To address these challenges, researchers turn to dynamic behavior analysis methods, monitoring interactions between executed code and a virtual environment. However, these detection methods can be resource-intensive. Machine learning proves invaluable for scrutinizing the behavior of any process or application [16]. Emerging technologies, such as machine learning, present a novel research focus, particularly in the realm of ransomware detection, offering considerable potential for innovative solutions [17]. Leveraging Machine Learning (ML) methodologies facilitates the automated detection of malware [18], including ransomware, by analyzing their dynamic behaviors, thereby enhancing overall security [19]. Various algorithms, including Decision Tree (DT) [20], Random Forest (RF) [21], Naïve Bayes (NB) [22], Logistic Regression (LR) [23], and Neural Network (NN)-based architectures [24], exhibit promising efficacy in the classification and detection of ransomware [25]. This study undertakes a thorough evaluation, exploring machine learning techniques for ransomware classification Frequent updates to signatures and rules for detecting new variants may improve response times but expose systems to emerging threats. The encrypted nature of cloud data poses challenges for conventional detection methods, limiting accessibility and analysis [26], [27]. In response, transfer learning emerges as a technique allowing the transfer of knowledge from pre-trained models on extensive datasets, enhancing detection accuracy, even in the absence of labeled ransomware samples. This strategy addresses the scarcity of ransomware data, particularly in cloud environments, thereby strengthening detection capabilities [27], [28]. The impetus for this investigation arises from the escalating frequency and severity of ransomware attacks targeting cloud environments, rendering traditional detection methods ineffective against evolving ransomware strains. The encrypted nature of cloud data adds complexity, emphasizing the need for innovative approaches to identify ransomware attacks within this encrypted context [29]. The adoption of transfer learning and deep learning ensembles is motivated by the desire

for enhanced detection precision and adaptability. The study aims to improve detection capabilities by overcoming limitations associated with insufficient labeled data through transfer learning. The integration of deep learning ensembles into detection systems has the potential to enhance cloud-based security by capturing more features [14]. By utilizing state-of-the-art methods such as transfer learning and deep learning ensembles for ransomware detection [30], this research strives to fortify the defenses of cloud-based systems and safeguard sensitive data against potential cyber threats.

3. Method

3.1. Comparative Analysis of Ransomware Detection Algorithms in Python

In this study, an empirical investigation into the effectiveness of several machine learning algorithms for ransomware detection was conducted. Specifically, Long Short-Term Memory (LSTM), Random Forest, Logistic Regression, and Neural Network algorithms were implemented using Python scripts. The accuracy of each algorithm was meticulously assessed, and the results have been succinctly presented in Table 1. The tabulated outcomes illuminate a notable variance in performance among the employed algorithms. Remarkably, the analysis underscores the superior efficacy of one algorithm over the others. Consequently, a more in-depth scrutiny of this particular algorithm will be undertaken, accompanied by rigorous testing procedures to further substantiate its proficiency in the realm of ransomware detection.

Table.1 The Result Of Python Code Applied

Variable	Accuracy
Random Forest	0.8
Logistic Regression	0.4
LSTM	0.3
Neural network	0.2

3.2. Data Analyzing

Our graph features are tailored to measure distinct transaction patterns within the Bitcoin dataset. The "Loop" feature serves to quantify transactions that involve the splitting of coins, their movement through various paths in the network, and eventual merging in a single address. This final address is typically associated with selling and converting the coins to fiat currency. The "Weight" feature specifically assesses merge behavior by gauging whether a transaction has more input addresses than output addresses. It captures instances where coins in multiple addresses undergo a series of merging transactions, ultimately accumulating in a final address.

Similarly, the "Count" feature is crafted to quantify the merging pattern, focusing on the number of transactions involved. Lastly, the "Length" feature is designed to measure mixing rounds on Bitcoin. It assesses transactions that distribute similar amounts of coins through multiple rounds, utilizing newly created addresses to obscure the origin of the coins.

The TP Rate represents the proportion of actual ransomware instances correctly identified by the model. A higher TP Rate indicates that the algorithm is adept at correctly classifying instances as ransomware. The FP Rate is the proportion of non-ransomware instances incorrectly classified as ransomware. A lower FP Rate is desirable, as it indicates fewer false alarms or instances where benign data is mistakenly identified as ransomware. The PRC Area summarizes the precision-recall trade-off across different decision thresholds. A larger PRC Area indicates better performance, reflecting the model's ability to maintain high precision while achieving high recall. It's especially useful when dealing with imbalanced datasets, such as those with a small number of ransomware instances. Specific ransomware detection models as shown in Table 2.

Table.2 The TPR, FPR, and PRC Area for the Algorithms

Algorithm	TP Rate	FP Rate	PRC Area
Random Forest	0.913	0.970	0.776
Neural Network	0.756	0.877	0.816
Logistic regression	0.687	0.798	0.714

This metric is especially valuable in handling imbalanced datasets, where maintaining high precision and achieving high recall are both critical. With this understanding, let us delve into the nuanced evaluations of specific ransomware detection models as shown in Table 2: Random Forest, Neural Network, and Logistic Regression as follows:

- Random Forest: It demonstrates a high TP Rate, indicating strong ransomware detection capability. However, the FP Rate is also relatively high, suggesting that there might be some false positives. The PRC Area is moderate, showing a good balance between precision and recall.
- Neural Network: It achieves a good balance between TP Rate and FP Rate, with a higher PRC Area, suggesting effective ransomware detection. It shows a strong performance in capturing ransomware instances while minimizing false positives.
- Logistic Regression: While it has a relatively lower TP Rate, it also exhibits a lower FP Rate. The PRC Area is moderate, indicating reasonable performance. It may be a more conservative model, being cautious about classifying instances as ransomware.

3.3. Correlation Analysis

We computed and analyzed correlation matrices to understand the relationships between variables in our dataset. The correlation analysis provides insights into the linear dependencies and associations among the features, setting the foundation for interpreting the collaborative behavior of the algorithms. Correlation matrix as show in Fig. 3.

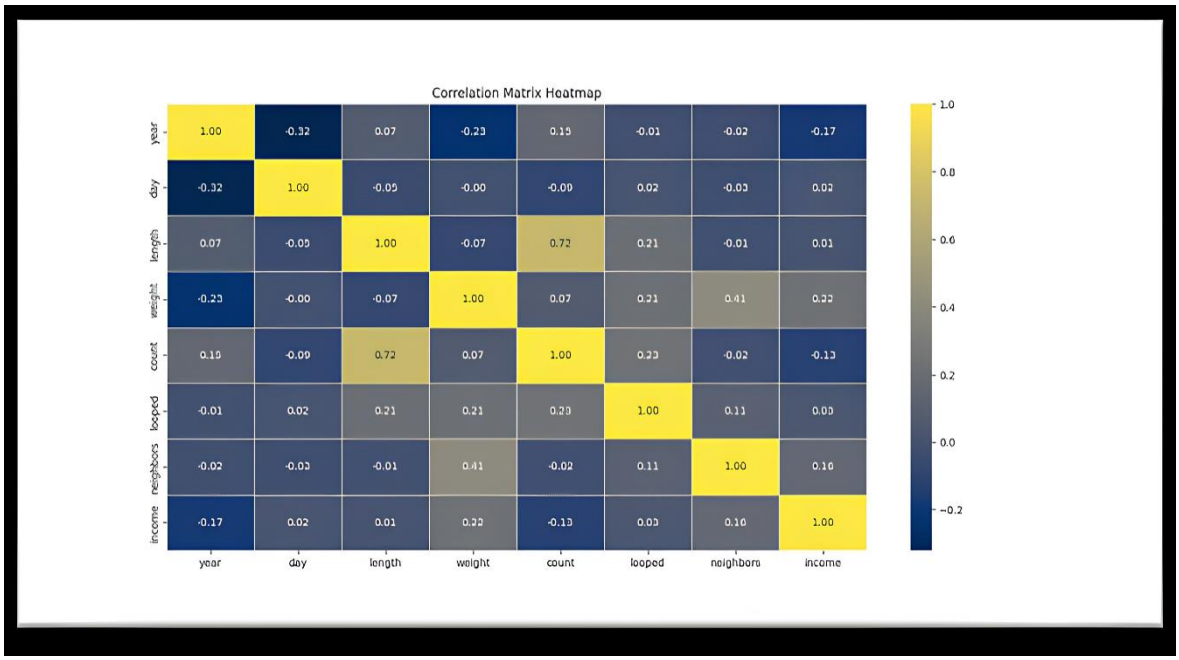

Fig. 3. Correlation matrix for dataset.

The correlation matrix functions as a crucial input for complex analyses such as exploratory factor analysis and structural equation models. Due to its symmetrical structure, half of the correlation coefficients in the matrix are redundant and serve no additional purpose. Essentially, a correlation matrix offers a concise summary of the relationships between all variables in a dataset. It is worth noting that the correlation coefficients along the diagonal of the matrix consistently equal 1, representing the perfect correlation of each variable with itself.

3.4. Scatter and Density Distribution

The direction and spread of points can indicate the correlation between variables. The patterns suggest relationships of the variable in our dataset. The diagonal contains kernel density estimates for each variable. It represents the distribution of values for each variable, Peaks and valleys in the diagonal plots indicate areas of higher or lower density in the data distribution and the areas of higher density suggest regions where the variables are more concentrated. Observe how the shape of the density plots off the diagonal corresponds to the scatter plots.

In Fig. 4 Each point in the scatter plots represents the relationship between two variables, and from that we can find the data points in Fig. 4 the data points in the graph are tightly close to each other. Consequently, the dataset displays a strong relationship between data.

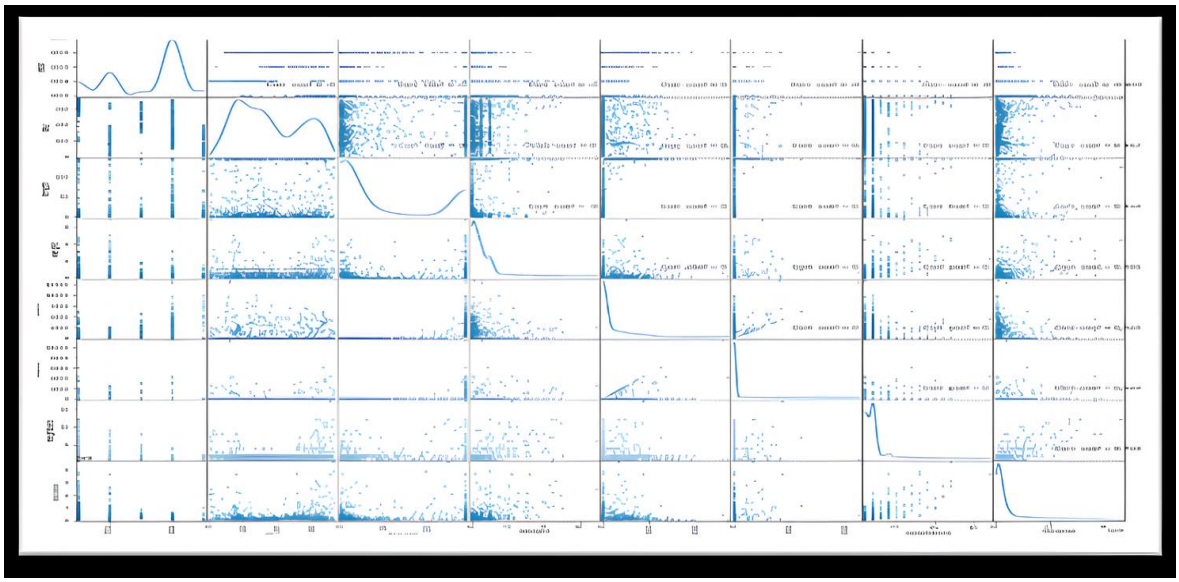

Fig. 4. Scatter and Density Plots

3.5. Column Distribution Analysis

We investigated the distribution of individual columns within the dataset, shedding light on the statistical properties of each feature. By employing Python's matplotlib and Seaborn libraries, alongside Weka's visualization capabilities, we gained a comprehensive understanding of the feature distributions. Fig. 5 show the structure of our data and to find anomalies that might be affecting the quality of it.

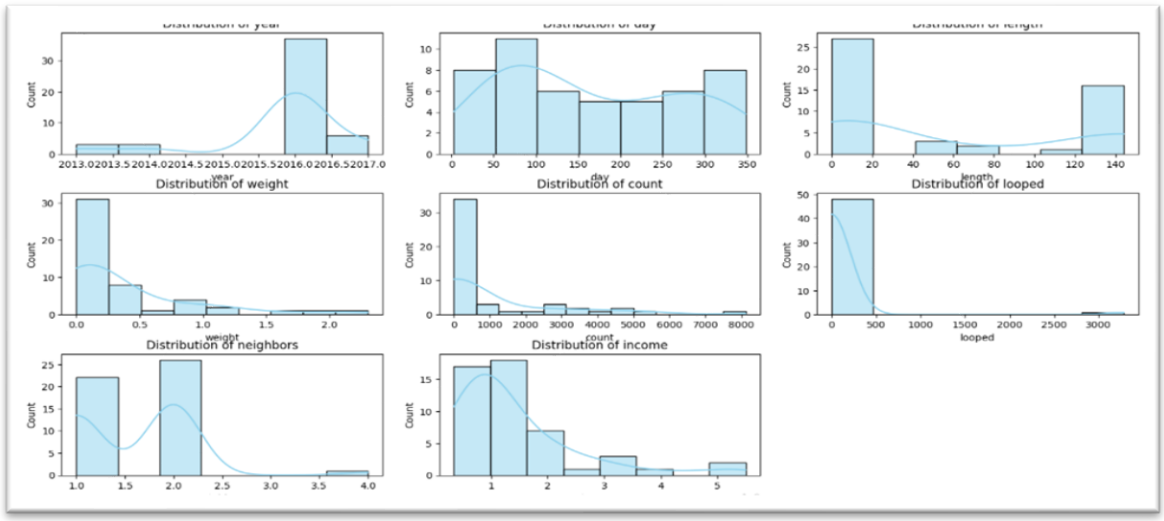

Fig. 5. Column Distribution Analysis

4. Conclusion

In this comprehensive study on ransomware detection, our exploration began with a thorough examination of the dataset. Leveraging exploratory data analysis (EDA) techniques, we delved into the distribution of critical variables, revealing nuanced insights into the characteristics of ransomware instances. The Scatter and Density Plot visualizations provided a dynamic portrayal of relationships between variables, offering a nuanced understanding of potential correlations and distributions. The use of machine learning algorithms, including Random Forest, Neural Network, and Logistic

Regression, was instrumental in evaluating the effectiveness of ransomware detection. The analysis of True Positive Rate, False Positive Rate, and Precision-Recall Curve Areas provided a robust quantitative assessment, highlighting the strengths and considerations of each algorithm. Notably, the Neural Network demonstrated a balanced performance, showcasing the potential for effective ransomware detection while minimizing false positives. As we navigated through the intricacies of our dataset, the importance of open-source software (OSS) became evident. OSS not only facilitated our analyses but also aligned with the principles of transparency, collaborative innovation, and flexibility, essential in the rapidly evolving landscape of cybersecurity. In conclusion, this study contributes valuable insights to the ongoing discourse on ransomware detection. By combining meticulous data analysis, visualization techniques, and machine learning evaluations, we have unveiled patterns and relationships crucial for effective detection strategies. As we confront the evolving threat of ransomware, this research stands as a testament to the power of interdisciplinary approaches, emphasizing the significance of robust data exploration and collaboration in cybersecurity.

References

[1] L. Caviglione *et al.*, "Tight Arms Race: Overview of Current Malware Threats and Trends in Their Detection," *IEEE Access*, vol. 9, pp. 5371–5396, 2021, doi: 10.1109/ACCESS.2020.3048319.

[2] K. Stoddart, "Non and Sub-State Actors: Cybercrime, Terrorism, and Hackers," in *Cyberwarfare*, Cham: Springer International Publishing, 2022, pp. 351–399, doi: 10.1007/978-3-030-97299-8_6.

[3] Y. K. Bin Mohamed Yunus and S. Bin Ngah, "Ransomware: stages, detection and evasion," in *2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM)*, Aug. 2021, pp. 227–231, doi: 10.1109/ICSECS52883.2021.00048.

[4] R. Moussaileb, N. Cuppens, J.-L. Lanet, and H. Le Bouder, "A Survey on Windows-based Ransomware Taxonomy and Detection Mechanisms," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–36, Jul. 2022, doi: 10.1145/3453153.

[5] G. Hull, H. John, and B. Arief, "Ransomware deployment methods and analysis: views from a predictive model and human responses," *Crime Sci.*, vol. 8, no. 1, p. 2, Dec. 2019, doi: 10.1186/s40163-019-0097-9.

[6] S. H. Kok, A. Abdullah, N. Z. Jhanjhi, and M. Supramaniam, "Ransomware, Threat and Detection Techniques: A Review," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 2, pp. 136–146, 2019. [Online]. Available at: http://paper.ijcsns.org/07_book/201902/20190217.pdf.

[7] A. Rahimian, L. Nouh, D. Mouheb, and H. Huang, *Binary Code Fingerprinting for Cybersecurity*. p. 249, 2020. [Online]. Available at: https://link.springer.com/book/10.1007/978-3-030-34238-8.

[8] A. Kapoor, A. Gupta, R. Gupta, S. Tanwar, G. Sharma, and I. E. Davidson, "Ransomware Detection, Avoidance, and Mitigation Scheme: A Review and Future Directions," *Sustainability*, vol. 14, no. 1, p. 8, Dec. 2021, doi: 10.3390/su14010008.

[9] O. M. K. Alhawi, J. Baldwin, and A. Dehghantanha, "Leveraging Machine Learning Techniques for Windows Ransomware Network Traffic Detection," 2018, pp. 93–106, doi: 10.1007/978-3-319-73951-9_5.

[10] S. Homayoun *et al.*, "DRTHIS: Deep ransomware threat hunting and intelligence system at the fog layer," *Futur. Gener. Comput. Syst.*, vol. 90, pp. 94–104, Jan. 2019, doi: 10.1016/j.future.2018.07.045.

[11] T. McIntosh, A. S. M. Kayes, Y.-P. P. Chen, A. Ng, and P. Watters, "Ransomware Mitigation in the Modern Era: A Comprehensive Review, Research Challenges, and Future Directions," *ACM Comput. Surv.*, vol. 54, no. 9, pp. 1–36, Dec. 2022, doi: 10.1145/3479393.

[12] S. Kalhoro, M. Rehman, V. Ponnusamy, and F. B. Shaikh, "Extracting Key Factors of Cyber Hygiene Behaviour Among Software Engineers: A Systematic Literature Review," *IEEE Access*, vol. 9, pp. 99339–99363, 2021, doi: 10.1109/ACCESS.2021.3097144.

[13] K. Maennel, S. Mäses, and O. Maennel, "Cyber Hygiene: The Big Picture," in *Lecture Notes in Computer Science*, 2018, pp. 291–305, doi: 10.1007/978-3-030-03638-6_18.

[14] A. Vehabovic, N. Ghani, E. Bou-Harb, J. Crichigno, and A. Yayimli, "Ransomware Detection and Classification Strategies," in *2022 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, Jun. 2022, pp. 316–324, doi: 10.1109/BlackSeaCom54372.2022.9858296.

[15] G. Ramesh and A. Menen, "Automated dynamic approach for detecting ransomware using finite-state machine," *Decis. Support Syst.*, vol. 138, p. 113400, Nov. 2020, doi: 10.1016/j.dss.2020.113400.

[16] A. Alraizza and A. Algarni, "Ransomware Detection Using Machine Learning: A Survey," *Big Data Cogn. Comput.*, vol. 7, no. 3, p. 143, Aug. 2023, doi: 10.3390/bdcc7030143.

[17] D. W. Fernando, N. Komninos, and T. Chen, "A Study on the Evolution of Ransomware Detection Using Machine Learning and Deep Learning Techniques," *IoT*, vol. 1, no. 2, pp. 551–604, Dec. 2020, doi: 10.3390/iot1020030.

[18] A. Brown, M. Gupta, and M. Abdelsalam, "Automated machine learning for deep learning based malware detection," *Comput. Secur.*, vol. 137, p. 103582, Feb. 2024, doi: 10.1016/j.cose.2023.103582.

[19] F. Noorbehbahani and M. Saberi, "Ransomware Detection with Semi-Supervised Learning," in *2020 10th International Conference on Computer and Knowledge Engineering (ICCKE)*, Oct. 2020, pp. 024–029, doi: 10.1109/ICCKE50421.2020.9303689.

[20] F. Ullah *et al.*, "Modified Decision Tree Technique for Ransomware Detection at Runtime through API Calls," *Sci. Program.*, vol. 2020, pp. 1–10, Aug. 2020, doi: 10.1155/2020/8845833.

[21] B. M. Khammas, "Ransomware Detection using Random Forest Technique," *ICT Express*, vol. 6, no. 4, pp. 325–331, Dec. 2020, doi: 10.1016/j.icte.2020.11.001.

[22] B. Ramadhan, Y. Purwanto, and M. F. Ruriawan, "Forensic Malware Identification Using Naive Bayes Method," in *2020 International Conference on Information Technology Systems and Innovation (ICITSI)*, Oct. 2020, pp. 1–7, doi: 10.1109/ICITSI50517.2020.9264959.

[23] Z. Akram, M. Majid, and S. Habib, "A Systematic Literature Review: Usage of Logistic Regression for Malware Detection," in *2021 International Conference on Innovative Computing (ICIC)*, Nov. 2021, pp. 1–8, doi: 10.1109/ICIC53490.2021.9693035.

[24] H. Madani, N. Ouerdi, A. Boumesaoud, and A. Azizi, "Classification of ransomware using different types of neural networks," *Sci. Rep.*, vol. 12, no. 1, p. 4770, Mar. 2022, doi: 10.1038/s41598-022-08504-6.

[25] L. Chen, C.-Y. Yang, A. Paul, and R. Sahita, "Towards resilient machine learning for ransomware detection," no. Ml, p. 10, 2018. [Online]. Available at: https://arxiv.org/abs/1812.09400.

[26] M. Abdullah Alohali, M. Elsadig, F. N. Al-Wesabi, M. Al Duhayyim, A. Mustafa Hilal, and A. Motwakel, "Optimal Deep Learning Based Ransomware Detection and Classification in the Internet of Things Environment," *Comput. Syst. Sci. Eng.*, vol. 46, no. 3, pp. 3087–3102, 2023, doi: 10.32604/csse.2023.036802.

[27] K. Lee, S.-Y. Lee, and K. Yim, "Machine Learning Based File Entropy Analysis for Ransomware Detection in Backup Systems," *IEEE Access*, vol. 7, pp. 110205–110215, 2019, doi: 10.1109/ACCESS.2019.2931136.

[28] O. Aslan and A. A. Yilmaz, "A New Malware Classification Framework Based on Deep Learning Algorithms," *IEEE Access*, vol. 9, pp. 87936–87951, 2021, doi: 10.1109/ACCESS.2021.3089586.

[29] S. Il Bae, G. Bin Lee, and E. G. Im, "Ransomware detection using machine learning algorithms," *Concurr. Comput. Pract. Exp.*, vol. 32, no. 18, Sep. 2020, doi: 10.1002/cpe.5422.

[30] G. Apruzzese *et al.*, "The Role of Machine Learning in Cybersecurity," *Digit. Threat. Res. Pract.*, vol. 4, no. 1, pp. 1–38, Mar. 2023, doi: 10.1145/3545574.