Wasit Journal for Pure Science

Journal Homepage: https://wjps.uowasit.edu.iq/index.php/wjps/index
e-ISSN: 2790-5241 p-ISSN: 2790-5233

An Approach to Android Ransomware Detection Using Deep Learning

Rasha Hani Salman[1]*, Rawaa Ismael Farhan[3]

[1]Second Author Affiliation,
College of Education for Pure Science College, Wasit University, IRAQ

[2]Second Author Affiliation
College of Education for Pure Science College, Wasit University, IRAQ

*Corresponding Author: Rasha Hani Salman

DOI: https://doi.org/ 10.31185/wjps.325
Received 01 December 2023; Accepted 22 February 2024; Available online 30 March 2024

ABSTRACT: As mobile devices continue to grow in popularity, the threat of ransomware attacks on Android devices has escalated, leading to serious privacy and financial risks for users. Traditional methods of ransomware detection have become less effective due to the evolving nature of these attacks. Consequently, the application of deep learning techniques offers promising potential for the detection and prevention of Android ransomware. This paper proposes an approach that utilizes deep learning algorithms to analyze and identify ransomware behavior on Android devices, aiming to enhance the security measures against these malicious threats. The model we used is a feedforward neural network model using the Keras Sequential. The model consists of three layers of densely connected neurons. The first layer has 64 units, the second layer also has 64 units, and the third and final layer has 2 units. The proposed model gave accuracy 98.9%.

Keywords: ransomware, android devices, deep learning, malicious threats.

1. INTRODUCTION

With the rapid growth of the Android operating system and its extensive user base, it has become an attractive target for cybercriminal activities. One of the most prevalent threats to Android users in recent years is the rise of Android ransomware attacks. These malicious attacks involve the unauthorized encryption of user's data, rendering it inaccessible until a ransom is paid to the attackers.[1]

Android ransomware attacks have witnessed a significant surge in both frequency and sophistication, posing severe threats to individuals, businesses, and even governmental agencies. The attackers exploit various means, such as phishing emails, malicious apps, or drive-by downloads, to gain access to a victim's mobile device [1]. Once infected, ransomware encrypts critical files or locks the device entirely, leaving users with limited options for data recovery, often forcing them to pay the ransom.

These attacks have far-reaching consequences, both financially and emotionally. Victims often face significant financial losses due to ransom payments, data recovery efforts, or potential reputational damage. Moreover, the emotional toll of having personal or sensitive data stolen can be devastating, impacting individuals' trust in digital systems and compromising their sense of security.[2]

The traditional approaches to detect and prevent ransomware attacks on Android devices typically rely on rule-based methods, signature-based detection, or behavior monitoring. While effective to some extent, these methods often struggle to keep up with the rapidly evolving malware landscape. Hackers continuously devise new techniques to evade detection, rendering traditional approaches less effective over time.[4]

In recent years, there has been a growing interest in applying deep learning techniques to enhance the detection and identification of Android ransomware. Deep learning, a subset of machine learning, leverages artificial neural networks to analyze large volumes of data and learn patterns, enabling the system to make accurate predictions and classifications [5]. The motivation behind utilizing deep learning techniques lies in their potential to identify malware with high accuracy, even in the presence of previously unseen ransomware variants. Through their inherent ability to learn from

complex and non-linear relationships within data, deep learning models can develop unmatched capabilities to detect sophisticated ransomware attacks at scale.[6]

2. ANDROID RANSOMWARE OVERVIEW

Ransomware is a type of malicious software that encrypts or locks a user's files or device and demands ransom payment to restore access. It is designed to extort money from victims by holding their data hostage. Once a device is infected, the ransomware encrypts the victim's files, making them inaccessible. The attacker then displays a ransom message, usually demanding payment in cryptocurrency, and provides instructions on how to make the payment. The victim is typically given a limited timeframe to make the payment, after which the ransom amount may increase or data may be permanently deleted.[7]

Ransomware possesses the subsequent features:[8]

- Encryption: Ransomware locks the victim's files with powerful encryption methods, rendering them unreadable without the decryption key.
- Demands for ransom: In return for the decryption key, attackers typically want payment in cryptocurrencies.
- Time constraints: A deadline is sometimes included in ransom communications, which puts pressure on the victim to make a rapid payment.
- Anonymous payments: To make it harder to track down the transactions, attackers frequently employ anonymous payment systems like Bitcoin.
- Social engineering: In order to access victims' computers, ransomware frequently deceives them into opening infected attachments or clicking on nefarious sites.

Working principles of ransomware:

- Delivery: Ransomware is typically delivered via infected email attachments, malicious websites, or drive-by downloads.
- Infection: Once the victim's device is compromised, the ransomware installs itself and starts encrypting files silently in the background.
- Encryption: Ransomware uses encryption algorithms to encrypt files, making them inaccessible without the decryption key.
- Ransom message: After encryption, the ransomware displays a message demanding payment and provides instructions on how to pay the ransom.
- Payment and decryption: If the victim pays the ransom, the attacker provides the decryption key to restore access to the encrypted files, although there is no guarantee that the attacker will fulfill their promise.

Common types and variants of Android ransomware:[9]

- Simplocker: Simplocker was one of the first types of ransomware detected on Android devices. It encrypts various file types and demands payment to decrypt them.
- LockerPin: LockerPin locks the victim's device by changing the PIN required to unlock it. It demands payment to unlock the device and restore access.
- DoubleLocker: DoubleLocker combines traditional ransomware techniques with the ability to change the device's PIN, making it even harder for victims to regain access to their devices.
- Police-themed ransomware: This variant displays a fake message indicating that the device has been locked by law enforcement due to illegal activities. It demands payment to unlock the device.
- Fusob: Fusob is a sophisticated strain of Android ransomware that not only encrypts files but also steals personal data from the victim's device.

3. DEEP LEARNING TECHNIQUES FOR ANDROID RANSOMWARE DETECTION

Deep learning is a subfield of machine learning that focuses on developing algorithms inspired by the structure and functioning of the human brain's neural networks. These algorithms employ multiple layers of artificial neurons to extract high-level patterns and features from complex data. Among the different neural network architectures, Multilayer Perceptron (MLP), Gated Recurrent Units (GRU), and Long Short-Term Memory (LSTM) networks have emerged as powerful tools for various tasks, including malware detection. [10]

Feature extraction plays a crucial role in ransomware detection as it helps identify specific characteristics and patterns that distinguish malicious behavior from benign activities. Traditional methods often require manual feature engineering, relying on expert knowledge to define relevant features. However, deep learning eliminates the need for manual feature extraction by automatically learning useful representations from raw data. This allows the models to capture both low-level and high-level features, making them highly effective in detecting complex and evolving ransomware attacks.[11]

Deep learning algorithms have been successfully implemented for ransomware detection on Android devices. By leveraging various deep learning architectures such as MLP, GRU, and LSTM, researchers have developed models capable of efficiently detecting and classifying ransomware based on behavioral patterns. These models analyze factors like file access, permission requests, network traffic, and system calls to identify potential threats. The flexibility and

adaptability of deep learning techniques make them well-suited for dynamic and continuously evolving ransomware attacks.[12].

4. DATA COLLECTION AND PREPROCESSING

The datasets used for the experiment were gathered from AndroZoo, which has benign samples, and RansomProber, which provides malicious (ransomware) samples [13]. The entire dataset, which was gathered via Android Package Kits (APKs), comprises 27,117 ransomware and 27,117 benign samples with 16 features including the target, as seen from fig.1.

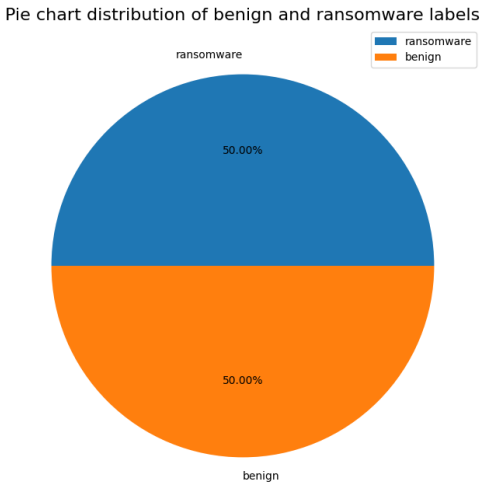

Figure 1. - Distribution of benign and ransomware samples

Since outliers—corrupted, distorted, or uninterpretable values—and missing data—unavailable values—are frequently found in datasets drawn from real-world sources, we performed feature engineering to transform the raw data we used for the study into a high-quality and practical format. mastery of the Python programming language. This stage involves using a variety of methods to find outliers, skewed features, redundant features, and missing values. The dataset was divided into 80% for training and 20% for testing.

5. MODEL ARCHITECTURE AND TRAINING

The model we used is a feedforward neural network model using the Keras Sequential. Where a feedforward neural is a type of artificial neural network model that is often used for supervised learning tasks. It consists of an input layer, one or more hidden layers, and an output layer, and Keras is a popular deep learning library that provides high-level abstractions for building neural networks.

The model consists of three layers of densely connected neurons. The first layer has 64 units and uses the rectified linear unit (ReLU) activation function. It takes an input of dimension 15.

The second layer also has 64 units and uses the ReLU activation function.

The third and final layer has 2 units and uses the softmax activation function. This is commonly used in multiclass classification problems to output probability distribution over the classes.

After building the layers, the model is compiled. The loss function used is "sparse_categorical_crossentropy," which is suitable for multiclass classification problems. The optimizer used is "adam."

Finally, the model is trained along with the number of epochs (200), batch size (2000), and a verbose level of 2 to display progress updates during training.

6. EVALUATION AND PERFORMANCE METRICS

Evaluation and performance metrics are essential in analyzing the effectiveness of machine learning models and classifiers. Accuracy refers to the proportion of correctly classified instances out of the total number of instances. It provides a general overview of the model's predictive power, but it might not be sufficient for imbalanced datasets. In such cases, other metrics like precision, recall, and the F1-score are crucial. Precision represents the ability of the model to correctly classify positive instances, measuring the fraction of true-positive predictions. Recall, also known as sensitivity or true positive rate, highlights the model's ability to identify all positive instances without missing any. The F1-score is a harmonic mean of precision and recall, offering a balanced evaluation metric when both types of errors (false positives and false negatives) are of equal concern. The performance of a model can also be represented by a confusion matrix, which visually displays the number of true positive, true negative, false positive, and false negative

predictions, enabling a deeper understanding of the model's performance. Overall, using a combination of accuracy, precision, recall, F1-score, and confusion matrix provides a comprehensive evaluation of machine learning models.

7. RESULTS AND DISCUSSION

The proposed method MLP yielded remarkable results with an accuracy of 0.989. This implies that the model successfully classified 98.9% of the instances accurately. In addition, the f1-score of 0.662 indicates a good balance between precision and recall. This indicates that the model achieved a decent level of accuracy in identifying positive instances while minimizing false positives and false negatives. The precision value of 0.5 denotes that out of all the instances identified as positive by the model, 50% were indeed positive. Lastly, with a recall value of 1, it indicates that the model identified all the actual positive instances in the dataset. And fig. 2 shows the confusion matrix of the predicted values, where TP=5344, FP=77, TN=5325, and FN=101. Overall, these results demonstrate the efficacy of our proposed MLP method in accurately classifying instances with high precision and recall rates.

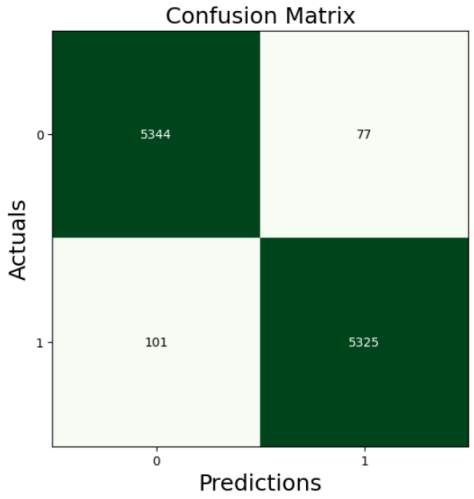

Figure 2. - Confusion matrix

To compare our results with the others, there have been numerous studies conducted on using machine learning techniques in the field of security. In one study, by W. Li et al. [14], a malware detection system was proposed that applies a deep learning algorithm. This system uses risky API calls and risk permission collections to implement a Deep Belief Network (DBN) model, allowing the automatic recognition of malware from malicious software. The authors utilized the DREBIN dataset to develop their scheme, achieving an accuracy of 90%.

R. Vinayakumar et al. [15] proposed a model based on both static and dynamic features to detect android malware. They utilized a type of recurrent neural network called Long Short Term Memory (LSTM) to analyze the long-term transient dynamics of characteristics with varying sequence lengths. The results demonstrated that the proposed model achieved good performance, with a 93.9% accuracy in dynamic features and a 97.3% accuracy in static features.

A. Naway and Y. Li [16] designed a malware detection model for android devices using the Autoencoder algorithm. Their model demonstrated high accuracy, with a recognition rate of 96.8% after considering five features.

In another study by S. Hou et al. [17], an auto-classification system for malware was proposed, utilizing two deep learning algorithms: deep belief networks and stacked autoencoders. These models rely on API calls to detect newly unknown malware that may impact android devices. The results showed a high accuracy rate of up to 96.66%.

8. FUTURE DIRECTIONS AND CHALLENGES

In the realm of deep learning-based ransomware detection, there are several potential future enhancements that can be explored. One avenue for improvement is the use of more complex neural network architectures, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs). These models have been successful in various domains and may offer improved performance in ransomware detection tasks. Additionally, incorporating more diverse features, such as system call sequences or network traffic patterns, can provide richer information for the model to learn from and potentially boost detection accuracy.

Ethical considerations and privacy concerns are of utmost importance in developing ransomware detection methods. While machine learning algorithms have the potential to mitigate ransomware threats, they also require access to sensitive user data. Therefore, it is crucial to carefully handle and protect user privacy during the training and deployment of these models. Striking a balance between effective detection and safeguarding user privacy is essential to ensure the ethical use of these technologies.

9. CONCLUSION

In conclusion, our proposed method for ransomware detection using a Multilayer Perceptron (MLP) has achieved impressive results, with an accuracy of 0.989, an F1-score of 0.662, a precision of 0.5, and a recall of 1. The advantages of using an MLP lie in its simplicity and ability to handle nonlinear relationships in the data. By leveraging deep learning techniques, we have made a significant contribution to the field of Android ransomware detection and demonstrated the potential for machine learning to combat this growing threat.

Moving forward, further research in this area is highly encouraged. Exploring more advanced neural network architectures, such as CNNs and RNNs, and incorporating additional features could yield even higher detection accuracy. Additionally, focusing on refining the ethical considerations and privacy concerns associated with ransomware detection methods will ensure the responsible and secure deployment of these technologies. By continuing to innovate and address these challenges, we can better defend against ransomware attacks and protect users' valuable data.

REFERENCES

[1] N. Scaife, H. Carter, P. Traynor and K. Butler, “Ransomware on Mobile Devices: The Threat and a Solution.,” In 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), pp. 393-400, IEEE, 2018.

[2] B. Kolosnjaji, M. Ramilli and D. Balzarotti, “Deep learning for classification of malware system-call sequences,” In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 1-12, ACM, 2017.

[3] S. Sharma, R. Krishna, and R. Kumar, “Android Ransomware Detection using Machine Learning Techniques: A Comparative Analysis on GPU and CPU,” Conference: 2020 21st International Arab Conference on Information Technology (ACIT), DOI:10.1109/ACIT50332.2020.9300108, pp. 1-6, 2020.

[4] A. Singh et al., “Enhancing Ransomware Attack Detection Using Transfer Learning and Deep Learning Ensemble Models on Cloud-Encrypted Data,” Electronics 2023, 12(18), DOI:10.3390/electronics12183899, 2023.

[5] Q. Xu, S. Zhang, S. Zhu and G. Wang, “ Neural network-based graphical for android malware detection,” IEEE Transactions on Information Forensics and Security, 11(3), pp. 472-483, 2016.

[6] H. Kim, K. Kwon, T. Kim, K. Park and J. Hong, “Using convolutional neural network models for android malware detection,” Multimedia Tools and Applications, 78(5), pp. 5943-5956., 2019.

[7] L. Nitschke, S. S. Javadi and B. Stock, “Machine Learning for Detecting Ransomware on Android Devices, ” IEEE Access, 8, pp.111346-111359, 2020.

[8] P. Vervier, E. Quiring, F. Monrose and M. Chew, “Understanding Android Ransomware and How to Defend Against It,” In International Workshop on Privacy Guard, pp. 256-279, Springer, Cham, 2016.

[9] Symantec: https://www.symantec.com/content/dam/symantec/docs/reports/threat-report/istr-24-2019-en.pdf, 2019.

[10] S. Amin, H. Nguyen, S. Abdullah, N. Naeem and X. Zhu, “Deep Learning Approaches for Ransomware Detection on Android Devices,” In IEEE Access, vol. 8, pp. 19174-19185, 2020.

[11] S. Singh, S. Sethi and K. Sharma, “A Comprehensive Study on Android Ransomware Detection Techniques: Traditional to Deep Learning,” In IEEE Access, vol. 8, pp. 44411-44439, 2020.

[12] M. Liu, Y. Wu and X. Zhang, “Detecting Android ransomware using recurrent neural networks,” In Proceedings of the 8th ACM Conference on Data and Application Security and Privacy, pp. 171-178, 2018.

[13] https://raw.githubusercontent.com/securycore/MLRD-Machine-Learning-Ransomware-Detection/master/data_file.csv, 2023.

[14] L. Wenjia, W. Zi, C. Juecong and C. Sihua, "An Android Malware Detection Approach Using Weight-Adjusted Deep Learning," in 2018 Int. Conf. Comput. Netw. Commun, 2018.

[15] R. Vinayakumar, k. S. Vinaya, P. Prabaharan and K. Sachin, "Detecting Android malware using Long Shortterm Memory (LSTM)," Journal of Intelligent & Fuzzy Systems, vol. 34, no. 4, pp. 1277-1288, 2018.

[16] N. Abdelmonim and L. Yuancheng, "Android Malware Detection Using Autoencoder," International Journal of Computer Engineering and Applications, 2019.

[17] H. Shifu, S. Aaron, C. Lingwei, Y. Yanfang and B. Thirimachos, "Deep Neural Networks for Automatic Android Malware Detection," 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 803-810, 2017.