Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 4 (2023)

A Hybrid Malware Vulnerability using Deep Learning Technique

ManojM [1], Dr.V.G.Rani [2]

[1]*Research Scholar, Department Of Computer Science, Sri Ramakrishna College Of Arts &Science for Women, Coimbatore, India.*

[2]*Associate Professor, Department Of Computer Science, Sri Ramakrishna College Of Arts & Science for Women, Coimbatore, India,*

Abstract - *Ransomware is on the rise and effective defensefrom it is of utmost importance to guarantee security of mobile users' data. Current solutions provided by antimalware vendors are signature-based and thus ineffective in removing ransomware and restoring the infected devices and files. To detect ransomware, the hybrid approach was developed using modified random forest and deep learning technique. The performance of our hybrid detection method is evaluated on a dataset that contains both ransomware and legitimate applications. Additionally, the performance of the static and of the dynamic stand-alone methods for comparison is evaluated. Results showed that detection method perform well in detecting ransomware, their combination in a form of a hybrid method performs best, being able to detect ransomware with 98% precision and having a false positive rate of less than 4%.*

Keywords: Ransomware, Hybrid detection, Random Forest, Deep Learning

I. Introduction

In recent years, the prevalence of malware has increased dramatically. In fact, ransomware has grown into one of the most prominent strains of cybercrime. We're seeing more cases of ransomware in 2017 than we have ever seen before due to its ability to autonomously propagate across the network. Clearly, ransomware mitigation techniques need to be designed in order to prevent successful attacks of malware. Luckily, there has been some work in the detection and mitigation of malware.

Ransomware is a software virus that holds a victim's files at ransom. Access to the files is not returned until a ransom is paid. There are two main types of ransomware in circulation today, crypto and locker-based ransomware. Crypto ransomware encrypts the files on a victim's computer and will only provide the decryption key for the files if a ransom is paid. On the other hand, locker ransomware leaves the victim's computer files intact but locks the user out of his or her computer, only returning access once a ransom is paid. Unfortunately, detecting various types of ransomware is an arduous task. Developing a long term solution to ransomware detection has proven difficult since ransomware developers are constantly updating their product to circumvent new detection techniques.

Broadly, ransomware is classified into two classes: Crypto ransomware and Locker ransomware. Crypto ransomware encrypts the files in victim's system, making the files impossible to use unless decrypted. Removing the ransomware or taking the hard drive to an unaffected system does not solve the problem as the victim does not have the key to decrypt the data. The victim is asked to pay the ransom specifically in the form of bitcoin to decrypt and recover the original files. Bitcoin is widely used by the attacker due to its anonymity as the identity of the attacker is difficult to trace. Paying ransom does not guarantee that the victim will get the decryption key to recover the data.

II. Related Work

Roseline, S.A et al [1] bypassed malware detection using evasion techniques such as code obfuscations, packing, etc., making detection methods ineffective. They proposed a diverse deep forest model for effective malware detection and classification. The system is focused on enhancing the existing malware detection systems in three aspects. Firstly, the PE binary files are converted to 2D grayscale images. Secondly, the images are processed in two phases, namely, sliding window scanning phase and cascade layering phase. The sliding window scanning phase is similar to convolutional neural networks where each pixel is processed using sliding windows, which allow considering critical features aiding for better prediction. The cascade layering phase consists of layers like deep learning but generates feature vectors without backpropagation.

Ullah, F et al [2] elaborated RW detection at runtime scheme which uses a preprocessed dataset that comprises benign and RW files. Benign is good ware, and RW is a special type of malware that keeps the data encrypted until a ransom is paid to the attacker. Their experimental outcomes demonstrate that the presented malware classification's testing and training accuracy is reached at 99.56%. Researchers stated some facts about sheltered device from attack and established some parameters to save data from the attack in the future, because RW is Trojan-type attack and malware, and so anomaly-based IDS may be used in the future for detecting abnormal behaviors of the network.

Xu, Z et al [3] introduced a framework for malware detection based on online analysis of virtual memory access patterns using machine learning. This framework was applied to the application-specific malware detection scenario which targets detecting malware infected runs of known applications. They addressed the challenge of online memory data collection using a system/function-call epoch based memory access summary. They experimentally covered both kernel and user level threats and demonstrated very high detection accuracy against kernel level rootkits and user level memory corruption attacks.

Mills et al [4] elaborated NODENS system has shown that it is possible to create lightweight, accurate and most importantly, interpretable automated malware detection systems. While the dataset used in the training and testing of both systems are comparatively small, the current accuracy of NODENS and proven ability to refit and identify previously undetected malware processes, shows that the system can remain at the forefront of malware detection and is highly adaptable, a key consideration in the ever changing threat landscape of cyber security.

The in-depth, decision specific interpretability that NODENS offers added weight to assessments drawn from analysis of raw data and makes it easier for future analytical work to be carried out, using easily understandable output formats. This will help increase awareness of both what a malware process is doing on a system and aid in the creation of a generalised malware model for further malware orientated research. Such knowledge will aid in the accuracy of antimalware systems, allowing both researchers and commercial vendors to hone in on malware specific signatures, such as the potentially unique memory foot print of malware processes identified through NODENS, making it harder for malicious agents to create undetectable malware.

The biggest limitation for NODENS is the size of the dataset used to date, which is small, consisting of 146 malware samples in total, including the training dataset. This can only be solved though continuous testing of the systems against malware samples or through bulk malware data collection. However it is the author's opinion that the number of process details generated per sample helps offset this limitation. Another limitation is that all of the training and testing malware samples were run in a virtualized environment, meaning that it was not possible to train the system against 'VM aware' malware. Whilst a more sophisticated environment could be developed to fool the malware via networking checks, a genuine physical networking environment is required to fully test against VM aware malware.

Apruzzese, G., et al [5] observed that existing approaches are based on classification criteria that are too rigid for the highly variable cyber security domain. The intuition is that by developing more flexible models it is possible to counter the manipulation of malicious samples. For this reason, we present an original method that

limits the impact of adversarial perturbations by leveraging the defensive distillation technique. They considered the random forest algorithm due to its superior performance in cyber security detection tasks. An extensive campaign of experimental evaluations demonstrates the effectiveness of the proposed method, which achieves a twofold advantage over the state-of-the-art: in scenarios subject to adversarially manipulated inputs, it improves the detection rate up to 250%; in scenarios that are not subject to adversarial attacks, it achieves a similar or superior accuracy than existing techniques.

This latter achievement is of particular importance because existing approaches that aim to counter adversarial attacks are often subject to a reduced performance in non-adversarial settings. Despite these promising results, their method presents room for further improvements. The proposed approach represents an original contribution to design robust detectors with high detection rates and strong enough against adversarial attacks.

Taheri, R et al [6] proposed five different attack algorithms: a trivial algorithm, a benign distribution, KNN, LR, and a bio-inspired method based on the ant colony algorithm. They compared these algorithms with the most recent static approach based on a Jacobian method, called JSMA, in terms of providing adversarial examples based on Android mobile data to fool classification algorithms. We also propose two defense algorithms based on adversarial training and GAN architecture. We validate our attack and defense algorithms using three public datasets, namely the Drebin, Genome, and Contagio datasets, using API, intent and permission file types. We test our models before and after implementing attacks on three classification algorithms: the RF, SVM and Bagging algorithms. It is observed that using 300 ranked syntax features of these Android mobile datasets, the benign distribution and LR attack algorithms could fool the classification algorithms using the Drebin dataset.

Qiu, J et al [7] elaborated DL, as a revolutionary paradigm of ML, is significantly and rapidly influencing data-driven research due to its advanced characterization and high-level abstraction of data. The application of DL for Android malware detection or classification is a hot developing research topic with many unsolved challenges. The fast advancements of new DL algorithms and architecture will keep driving the enhancement of Android malware detection, classification, and analysis. Author surveyed many existing studies that employed DL methods for Android malware detection or classification. Based on our observations, there is an evolving trend of various neural network architectures being adopted for faithful representation and characterization of Android malware to capture the intrinsic semantic patterns of Android malware for improving the performance of the detection or classification tasks. They see several important lines of future research, including large curated benchmark datasets, domain-specific model architectures, and pre-trained models with wide applicability.

Machine Learning (ML) has been recognized for its potential for Android malware detection and classification. Although many ML methods have been proposed, human intelligence still plays a critical role in the design of different tools and approaches. In particular, the traditional ML-based Android malware analysis methods are heavily dependent on the hand-crafted features [2]. Such features reflect what cyber security experts consider as the most significant intrinsic characteristics of Android malware. However, the feature engineering process is time-expensive and errorprone, and the produced hand-crafted features are task-specific and often subjective to individual judgment [3, 4]. Traditional ML has found it particularly effective in Android malware detection. However, most of the ML-based methods require security experts to manually define the features to characterize Android malware, which heavily relies on experts' experience, level of expertise, and the depth of domain knowledge [5, 6]. It is a critical but open issue to generate generalizable representations of Android malware less dependable on human experts. Deep Learning (DL) has been proposed for Android malware detection to overcome the obstacles of feature engineering, with the expectation of replicating DL's success in image classification [7], machine translation [8, 9], and text classification [10]. The benefits of using DL are apparent: The layered structure of DL-based models enhances the learning of abstract and highly non-linear patterns, which helps capture the intrinsic characteristics of complex data [11, 12]; the deep neural network structures allow the features to be learned automatically and with multiple levels of abstraction [13, 14], which contributes to an improved level of generalizability and relieves the security experts from labor-intensive and possibly error-

prone feature engineering tasks [15–20]; and the DL approaches can help identify the latent features that a security expert might have never considered [21–27], which significantly expands the feature representation space. Specifically, the DL methods are often more suitable to capture the semantic knowledge within Android apps than the traditional ML methods, particularly when sufficient data exist to learn a meaningful semantic embedding. For instance, the dynamic behavioral logs generated by the Android apps or the source code of apps can be seen as a special type of language so the Natural Language Processing (NLP) involved DL techniques can be readily used to address Android malware detection issues.

III. Methodology

3.1 Approach and considerations to construct a Hybrid Approach

Random fuzzy forests are collections of decision trees based on fuzzy logic. We tested fuzzy random forests after being encouraged by the good results of random forests. The Fuzzy Random Forest categorization module employs numerous fuzzy decision trees to reap the benefits of both decision trees and fuzzy logic. The label information from the several leaves reached in each tree was then merged.

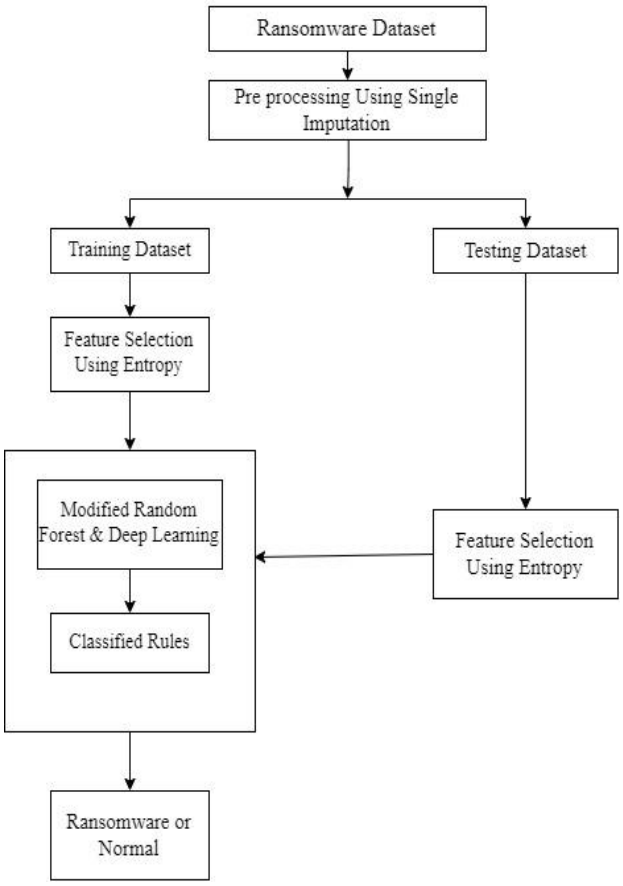

Fig 3.1 Proposed Methodology

In this work, Proposed hybrid detection method that is composed of a static method, to be used when applications are installed and/or updated, and a dynamic method, to be used at runtime. Static methods detect ransomware by considering features that can be obtained without running the applications.

Pseudo Code for Hybrid Random Forest and Deep Learning

Hybrid Random Forest and Deep Learning

Step 1 : Upload Ransomware dataset

Step 2 : Pre-processing is done using mean-imputation method

Step 3 : Modified Ransomware dataset is achieve

Random Forest

Select randomized data samples for calculating information gain and entropy.

Step 1 : Calculate entropy using

$E(S) = \sum_{i=1}^{c} p\,i \log 2pi$

$= -\frac{1}{N}\sum_{i=1}^{l}\sum_{j=1}^{N}[\mu_{ij} log_2\mu_{ij} + (1-\mu_{ij})log_2(1-\mu_{ij})]$- (3.1)

where S- for a given segment of data (*S*)

c-number of different class levels

p i - proportion of values falling into the class level *i*

N – number of instances

l - number of classes

Step 2 : Calculate information gain

Information

$Gain(A) = E(Currentset) - \sum E(allchildsets) -$ (3.2)

Where E is entropy of corresponding set

Step 3 : Attribute with maximum information gain is considered as root node

Step 4 : Sub tree is created based on root node.

Step 5 : If Sub tree has more attributes then

Generate lower level sub tree

Based on next entropy value

Repeat Step 5 till the sub tree has no more attributes left

End if

Deep Learning Technique

- Random Forest Decision Tree has to be initialized input layer of the network. The layers of neurons connect in a graph so that the signal passes in one direction.
- MLPs compute the input with the weights that exist between the input layer and the hidden layers.
- **Forward Propagation in MLP**

  Calculate the activation unit al(h) of the hidden layer

$$Z_1^{(h)} = a_0^{(in)} w_{0,1}^{(h)} + a_1^{(in)} w_{1,1}^{(h)} + \cdots + a_m^{(in)} w_{m,1}^{(h)}$$

$$a_1^h = \emptyset(z_1^h)$$

- Activation unit is the result of applying an activation function φ to the z value. It must be differentiable to be able to learn weights using gradient descent. The activation function φ is often the sigmoid (logistic) function

$$\emptyset(z) = \frac{1}{1 + e^{-z}}$$

- MLPs use activation functions to determine which nodes to fire. Activation functions include sigmoid functions.
- MLPs train the model to understand the correlation and learn the dependencies between the independent and the target variables from a training data set.
- output layer:

Z(out) = A(h) W(out)=A(out)

Dynamic methods, instead, are based on features that can only be obtained at runtime and that represent the behavior of applications. Static approaches are less computationally intense than dynamic methods and that they do not need applications to be run for identifying malware, but they are typically ineffective with obfuscated code as well as with run-time infections. On the other hand, dynamic methods areeffective in identifying new threats, outperforming static methods, but they need applications to be run to identify malicious behaviour, potentially infecting the device

IV. Results And Discussion

Evaluation metrics

1. Recall: The number of correct positive predictions among all the positive samples.

Mathematically:

$Recall = TP/TP + FN$

Where, TP is True Positive (quantity of correct positive predictions) and FN is False Negative (quantity of misclassified positive predictions)

2. Precision:

The proportion of the correctly identified positives to all the predicted positives.

Mathematically:

$Precision = TP/(TP + FP)$

3. $F1\ score$: The harmonic means of Precision and Recall.

$F1\ score$ is a better performance metric than the accuracy metric for imbalanced data.

$F1 = 2 \times Precision \times Recall/ (Precision + Recall)$

The F-beta score is the weighted harmonic mean of precision of recall where F-beta value at 1 means perfect score (perfect precision and recall) and 0 is worst.

$F\beta = (1 + \beta^2)\ Precision \times Recall/ (\beta^2 \times Precision) + Recall$

Fig 4.1 Precision Comparison of Random Forest and Fuzzy Random Forest

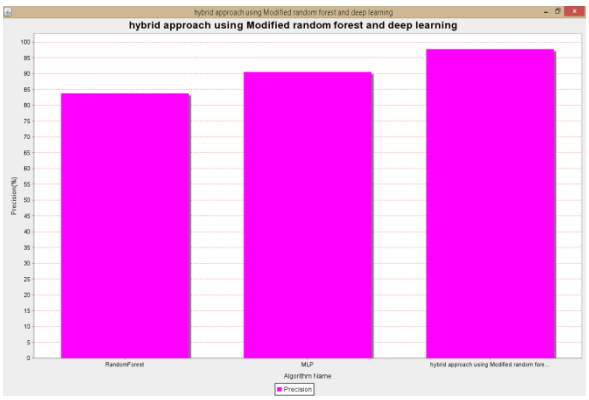

Fig 4.2 Performance Metrics Comparison of Random Forest and Fuzzy Random Forest

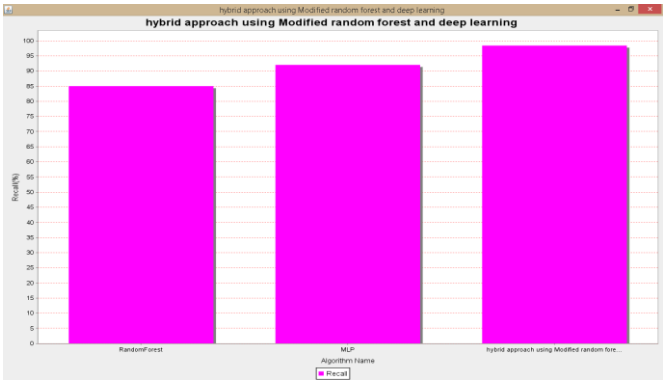

Fig 4.3 Performance Comparison of Random Forest and Fuzzy Random Forest

TABLE I COMPARISON OF PERFORMANCE METRICS

Technique/Metrics	Precision	Recall	Accuracy
Hybrid Approach	97.8	98.4	98
MLP	90.56	92.01	91
Random Forest	83.8	85	84.4

The above figure shows the comparison of performance metrics of precision, recall and accuracy of random forest and fuzzy random forest malware classification techniques. It is clearly depicted that fuzzy based random forest classification yields better results than random forest in ransomware classification.

V. Conclusion

In this research work, a hybrid approach to ransomware detection that has a 98% detection rate coupled with a false positive rate below 4%, even when analyzing previously unseen applications was proposed. This performance was achieved using dynamic method to complement the static one, thus increasing coverage and

allowing us to put together the advantages of both methods. Finally, given high achieved detection accuracy on one hand and the detection methods of low complexity used for on-device dynamic detection on the other, foresee that such hybrid method can be used to detect ransomware not only in mobile phones but also in other IoT devices.

References

[1] Roseline, S.A., Geetha, S., Kadry, S. and Nam, Y., 2020. Intelligent vision-based malware detection and classification using deep random forest paradigm. *IEEE Access*, *8*, pp.206303-206324.

[2] Ullah, F., Javaid, Q., Salam, A., Ahmad, M., Sarwar, N., Shah, D. and Abrar, M., 2020. Modified decision tree technique for ransomware detection at runtime through API calls. *Scientific Programming*, *2020*.

[3] Xu, Z., Ray, S., Subramanyan, P. and Malik, S., 2017, March. Malware detection using machine learning based analysis of virtual memory access patterns. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017* (pp. 169-174). IEEE.

[4] Mills, A., Spyridopoulos, T. and Legg, P., 2019, June. Efficient and interpretable real-time malware detection using random-forest. In *2019 International conference on cyber situational awareness, data analytics and assessment (Cyber SA)* (pp. 1-8). IEEE.

[5] Apruzzese, G., Andreolini, M., Colajanni, M. and Marchetti, M., 2020. Hardening random forest cyber detectors against adversarial attacks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *4*(4), pp.427-439.

[6] Taheri, R., Javidan, R., Shojafar, M., Vinod, P. and Conti, M., 2020. Can machine learning model with static features be fooled: an adversarial machine learning approach. *Cluster computing*, *23*(4), pp.3233-3253.

[7] Qiu, J., Zhang, J., Luo, W., Pan, L., Nepal, S. and Xiang, Y., 2020. A survey of android malware detection with deep neural models. *ACM Computing Surveys (CSUR)*, *53*(6), pp.1-36.

[8] G Data CyberDefense AG. 2019. Mobile malware report—no let-up with Android malware. Retrieved from https:// www.gdatasoftware.com/news/2019/07/35228-mobile-malware-report-no-let-up-with-android-malware.

[9] Daniel Arp, Michael Spreitzenbarth, MalteHubner, Hugo Gascon, and KonradRieck. 2014. DREBIN: Effective and explainable detection of Android malware in your pocket. In Proceedings of the 21st Network and Distributed System Security Symposium (NDSS'14).

[10] Bae, S.I., Lee, G.B. and Im, E.G., 2020. Ransomware detection using machine learning algorithms. *Concurrency and Computation: Practice and Experience*, *32*(18), p.e5422.

[11] Sechel, S., 2019. A comparative assessment of obfuscated ransomware detection methods. *InformaticaEconomica*, *23*(2), pp.45-62.

[12] Aslan, Ö.A. and Samet, R., 2020. A comprehensive review on malware detection approaches. *IEEE Access*, *8*, pp.6249-6271.

[13] Ferrante, A., Malek, M., Martinelli, F., Mercaldo, F. and Milosevic, J., 2017, October. Extinguishing ransomware-a hybrid approach to android ransomware detection. In *International symposium on foundations and practice of security* (pp. 242-258). Springer, Cham.

[14] Hwang, J., Kim, J., Lee, S. and Kim, K., 2020. Two-stage ransomware detection using dynamic analysis and machine learning techniques. *Wireless Personal Communications*, *112*(4), pp.2597-2609.

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

[16] Ding, Y., Zhang, X., Hu, J. and Xu, W., 2020. Android malware detection method based on bytecode image. *Journal of Ambient Intelligence and Humanized Computing*, pp.1-10.