# Detecting A Crypto-mining Malware By Deep Learning Analysis

**Shahad Aljehani**[1†] **and Hatim Alsuwat** [1††],
S44380037@st.uqu.edu.sa Hssuwat@uqu.edu.sa
[1] Department of Computer Science, College of Computer and Information Systems, Umm Al-Qura University, Saudi Arabia

**Summary**
Crypto-mining malware (known as crypto-jacking) is a novel cyber-attack that exploits the victim's computing resources such as CPU and GPU to generate illegal cryptocurrency. The attacker get benefit from crypto-jacking by using someone else's mining hardware and their electricity power. This research focused on the possibility of detecting the potential crypto-mining malware in an environment by analyzing both static and dynamic approaches of deep learning. The Program Executable (PE) files were utilized with deep learning methods which are Long Short-Term Memory (LSTM). The finding revealed that LTSM outperformed both SVM and RF in static and dynamic approaches with percentage of 98% and 96%, respectively. Future studies will focus on detecting the malware using larger dataset to have more accurate and realistic results.

***Keywords:***
*Crypto-mining, Crypto-jacking, Cryptography, Deep Learning, Detection*

## 1. Introduction

In an era where everything is being digitized, it is important to investigate that the modern technologies are truly secured. Attackers can exploit any small security bug to gain unauthorized access or privileges on a user's computer. One of the most attacked field is cryptocurrency[1], which is a digital currency that is secured using cryptography mechanisms with the use of decentralized blockchain networks [1] [2].

Cryptocurrency has gained wide popularity among investors and entrepreneurs around the world since its release in 2009 [3]. Its applications were not limited to banking and financial institutions, but also many other sectors have conducted business using cryptocurrencies in their systems such as Healthcare [4] and Insurance [5] systems. However, not all cryptocurrencies are generated legally, some cybercriminals exploit users' resources to make illegal cryptocurrencies in a so called process crypto-jacking, hence the need of detecting such behavior is extremely mandatory [2].

In order to do so, Machine Learning (ML) algorithms are used to identify patterns and build analytical model [5]. Since ML algorithms used for detecting patterns and predicting future data model, they have been used in many different applications such as Image recognition [4], Medical diagnosis [7] and Cryptocurrency mining [1]. For that, these classifers are used to build and test models that detect crypto-mining malware activities and report on them. Particularly, ML methods are used alongside with deep learning (DL) – which is a part of the ML –, many complex and practical problems can be solved since it applied on a larger dataset making better decisions and solutions [6]. The DL has two types; static and dynamic analysis. The static method check for malware signatures and specific keys in an application without executing it while the dynamic method uses and executes a malware sample to record and detect its behavior. These two methods can be applied on both In-browser crypto-jacking (a malware that exploits the interactions of web pages on user's CPU) and Host-based crypto-jacking ( where a malware owner uses a victim's computer as a zombie computer) [1]. However, most of the related works focused on in-browser crypto-jacking [9] [10] [12] [13] [14], while only few address the host-based crypto-jacking [1][11]. Even fewer works tries to detect crypto-mining malware using both static and dynamic methods [1][9][10]. For that, this research covers the detection of host-based Crypto-jacking by using both static and dynamic analysis of multiple ML classifiers. The research investigated the question:

- How can deep learning techniques such as LTSM, CNN and Random Forest be used to detect the crypto-mining malware behavior in a host-based controlled environment?

Particularly, it discussed and analyzed these subsidiary research questions:

1- How can dynamic features such as system calls be used in detecting crypto-mining malware using LTSM and CNN performance classifiers?
2- How can static features such as opcodes be used in detecting crypto-mining malware using Random Forest classifier?
3- What are the challenges of detecting crypto-mining activities?

The basic contribution of this research is to practice the use of ML classifiers to detect a new and epidemic malware in a context of testing and analyzing the behavior using various techniques and features. The lack ( or few) of the previous studies about the research problem motivated

Manuscript received June 5, 2022
Manuscript revised June 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.6.25

the author to conduct the study and further enhance the knowledge in the related field.

The structure of this paper is organized as this way; Section 2 introduces the background information, Section 3 explains the implied methodology, Section 4 shows the findings and Section 5 concludes the research and the future work.

## 2. Background

To have a comprehensive view about the research idea, several topics need to be identified and presented. Thus, this section will discussed these concepts.

### 2.1. Blockchain

Blockchain is a technology invented to allow the use of decentralization infrastructure for storing digital ledger of transactions in an immutable way [15]. This system - which also known as a Distributed Ledger Technology (DLT)- duplicates and distributes its data across network of users. It consists of a sequence of blocks that are chained to each other by using cryptography [16].

### 2.2. Cryptography

By its definition, cryptography is the study and practice of methods for secure communications between two parties. Both intended sender and receiver have keys in which they can decrypt an encrypted message [15]. One of the most and broad use of cryptography is the digital currencies; which known as cryptocurrency. The most known cryptocurrency is Bitcoin and it is the main reason that the cryptocurrency were invented [17].

### 2.3. Cryptocurrency

Cryptocurrency is a decentralized blockchain-based digital currency that uses a public and distributed ledger with techniques for privacy enabling that hide transactions from any observer. Basically, cryptocurrency is exist because of blockchain technology and cryptography mechanisms [13][18]. Moreover, cryptocurrency uses cryptography mechanisms for two main purposes; to have secured transactions and to authenticate these transfers. Beside Bitcoin, there are other cryptocurrencies that are well-known and broadly used namely, Monero, Tether and Ethereum, but Bitcoin still leads the market with Market cap: >US$775 billion in 2022 [19].

### 2.4. Cryptocurrency Mining

Generating cryptocurrency is done through a legitimate operation called cryptocurrency mining. It needs expensive hardware such as GPUs and extensive supply power to generate a valid mining result [20]. In particular, the process consists of nodes or specialized hardware that validate transactions and solve complex mathematical problems and in turn, successful miners make profit and receive new cryptocurrency as a reward for their useful efforts. However, this mining process can be exploited by attackers to make profits without the need to buy expensive hardware through a malware called crypto-jacking [21][2].

### 2.5. Crypto-jacking

This threat has changed the cyber world as it considered one of the common and hardest malware to be detected [1]. It involves the use of users' hardware to mine for cryptocurrency in an illegal way. One of real-world crypto-jacking examples happened in June 2020, the well-known cyber security company called Palo Alto Networks had identified a crypto-jacking scheme that was placed within docker images on the Docker Hub network which can be accessed publicly. The estimations of illegal profits from this operation was $36,000 [22].

### 2.5.1. Platform Types

The attacker has two ways to exploit victim's computer, In-browser and Host-based methods.

#### 2.5.1.1. In-browser Crypto-jacking

Web technologies such as JavaScript and Web Assembly use user' CPU for computational purposes. These can act as entrance for many unauthorized accesses which illegal crypto miners can benefit from them. In-browser crypto-jacking happened when a user visits an infected website that has a malware script injected in it. Once the code is executed, the crypto mining process is started in a silent way, making victim's hardware runs complex mathematical problems [23][24]. Figure 1 below shows the lifecycle of an in-browser crypto-mining malware.

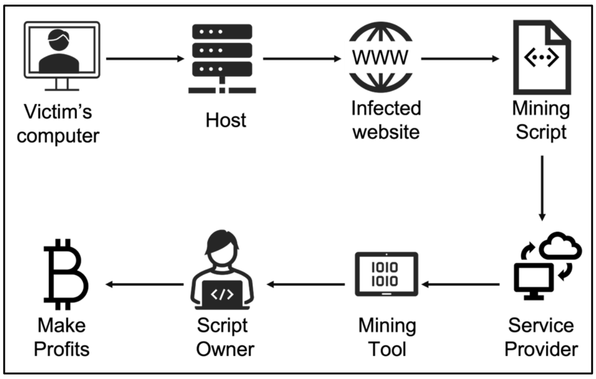

Fig. 1 An In-browser Crypto-jacking Malware

#### 2.5.1.2. Host-based Crypto-jacking

The other method of crypto-jacking is similar to phishing, which starts by an legitimate-looking email deliver to a victim that has an infected link. Once they click on the it, the crypto-mining script run on the victim's CPU and GPU as a background task with the user being unaware of it [24]. Figure 2 below shows the lifecycle of an host-based crypto-mining malware.

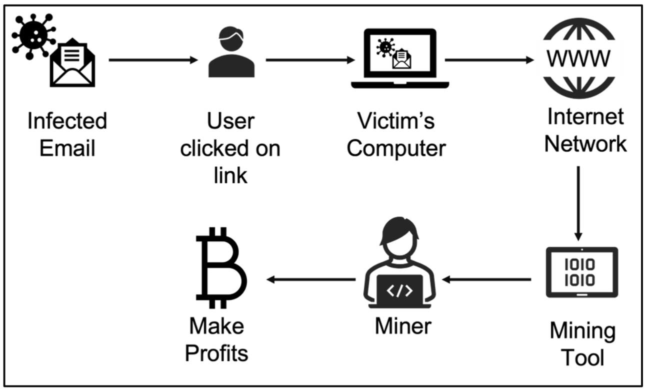

Fig. 2 A Host-based Crypto-jacking Malware

These two methods are usually combined together to increase the profits. Another way of maximizing the returns is sometimes the crypto-mining code comes in multiple versions to target multiple network infrastructure [24]. Therefore, looking for methods to detect this critical malware is mandatory.

### 2.6. Detection Methods and Their Analysis Features

There are two primary methods for detecting a malware; static and dynamic.

#### 2.6.1. Static Analysis

Static analysis is the process of detecting a crypto-mining by examining an application code to discover specific malware signatures and other defined keys. Due to its nature of non-execution technique, this method works well with various in-browser features such as Web-Assembly signatures and CPU cache events and host-based feature like System calls [25][26].

#### 2.6.2. Dynamic Analysis

Dynamic analysis requires the crypto-mining malware to be running in a controlled environment in which all activities of the malware are recorded for further behavioral observation. This type uses different analysis features such as Resource consumption, CPU usage and Opcodes for in-browser crypto-mining malware and CPU instructions, Network traffic and Packet sizes for host-based type of crypto-mining malware[27][28].

### 2.7. Machine Learning Classifiers

In order to analyzed the malware behavior. The collected data feature is used to build, train and test machine learning models and then measure their performance.

### 2.8. Related Work

To discuss how to detect crypto-mining malware, several related work are compared in terms of target crypto-mining type, analysis method, applied features, ML classifiers and the outcomes.

As a general view, the studies related to In-browser crypto-mining malware are more than the one that targeted the host-based type.

In [29], [26] and [30], an in-browser crypto-mining malware is detected using Web-Assembly signatures (WASM)feature. Both [26] and [30] analyzed it using a static method while [29] used a the dynamic method. The ML classifier in [29] showed a accuracy result of 98%. While others studies targeted the same in-browser type with dynamic analysis, they use different features such as CPU, Memory, Network behaviors [31], CPU usage [32][33][34] and Network traffic [35][28][36]. All of them showed an accuracy percentage that is over 90%.

On the other hand, studies that detected a host-based crypto-mining malware are relatively few. In [35], the author used the Packet sizes and Interarrival times to detect host-based malware with dynamic method. While in [37], they used System calls and opcode sequences and analyzed them by both static and dynamic methods.

As noted from the discussion of the related work, most of the studies focused on only one type of analysis method - which is dynamic -, while only few of them discussed the static method. Moreover, only one study have combined both Static and Dynamic analysis targeted a host-based crypto-mining malware [37]. Thus, this research addressed this gap and studied both Static and Dynamic analysis in a host-based crypto-mining malware with the use of System calls and opcode sequences to train and test multiple ML classifiers. This study applied the same approach used in [37] but with different and newer dataset to discover recent crypto-malware as well as find and study the patterns and behaviors of the newly discovered malwares. Table 1 below summarizes the related work and compares between them.

Table. 1 Summary of The Related Works

| Ref | Type | Method | Features | Classifier |
|---|---|---|---|---|
| [29] | B | D | WASM | Matching |
| [26] | B | S | WASM | SRSE |
| [30] | B | S | WASM, CPU cache events | Matching |
| [31] | B | D | CPU, Memory, Network traffics | CNN |
| [32] | B | D | CPU usage | MA |
| [33] | B | D | CPU usage | Matching |
| [34] | B | D | CPU usage, WASM execution time | CNN |
| [28] | B | D | Network packages | IL |
| [36] | B | D | Network traffic | DT |
| [35] | H | D | Network traffic | RF |
| [37] | B, H | S, D | System calls, opcode files | RNN, CNN |
| This study | B, H | S, D | System calls, opcode files | LTSM |

* SRSE: Symantec Rule Space Engine, CNN: Convolutional Neural Network, MA: Manual Analysis, RF: Random Forest, KFCV: k-Fold Cross Validation, IL: Incremental Learning, DT: Decision Tree, RNN: Recurrent Neural Network., D: dynamic, S: static, B: in-browser, H: Host-based.

## 3. Methodology

### 3.1. Dataset Preparation

This research has two dataset; opcodes and system calls used for static and dynamic analysis, respectively.

### 3.1.1. Dataset of Static Analysis (Opcodes)

Opcodes are Windows applications' operation codes, which are sets of machine learning instructions that define the operations that need to be performed by system calls. In the context of detecting a crypto-jacking malware, opcodes are used to monitor the requests between the OS kernel and the mining scripts.

The dataset was collected and downloaded from well-known repositories including VirusTotal as well as Virus Share. It contains two types of samples namely; malware and benign samples. For malware sample; it has more than 500 real-world cryptocurrency Malware in form of Portable Executable file and tagged as Crypto MS windows. While the benign sample consists of legitimate data from legal sources such as Microsoft Store and Coinmarketcap site. These data are the files of Crypto-minor and Crypto-wallet application with total number of 200 benign files. All malware data are executed on the MS windows platform [38]. The opcodes were extracted from the samples using IDA Pro.

IDA Pro is an integrated development environment for analyzing binary code by translating it to assembly language source code. This tool is used to extract the opcodes from the samples by disassembling them. The opcode analysis is a static analysis, which mean it does not requires any running or executing of any files. To train the models later, a sample of benign dataset is required. For that, Dynamic Link Library (DLL) of Microsoft Windows files were collected from a standard Windows installation [37]. Table 2 below shows the Opcode datasets of static analysis.

Table. 2 Opcodes Dataset

| Dataset | Number of sequences | Source |
|---|---|---|
| Op_ benign | 200 | Crypto-minor, Crypto-wallet, DLL. |
| Op_crypto | 500 | Crypto MS windows |

### 3.1.2. Dataset of Dynamic Analysis (System calls)

System calls are the APIs that allow connections between the OS kernel and the user applications. These calls run at level 0 which has a privilege of requesting any services from the kernel. In detecting a crypto-jacking malware, the frequent calls of certain libraries may considered a suspicious act and tagged as red flags, while the benign applications do not call these cryptographic libraries that much.

In this paper, these calls are needed for dynamic analysis by using the Cuckoo Sandbox tool. Then, the generated reports are used to train and test the deep learning classifiers. Table 3 below shows the system calls datasets of dynamic analysis.

Table. 3 System Calls Dataset

| Dataset | Number of sequences | Source |
|---|---|---|
| SysCall_ benign | 200 | PyWinMonkey |
| SysCall _crypto | 300 | WPE crypto-miner samples |

## 3.2. Environment Setup

In order to work on the malware dataset and analyze the files, an automated malware analysis system is used which is Cuckoo Sandbox. It is an open source environment that deeply analyze the behavior of any suspicious file in just a minute. It isolates the malware inside a mimic environment while executing the malware files to generate a detail report about it. As noted before, there are two types of deep analysis of the behavior of a cryptocurrency malwares; static and dynamic. The Cuckoo sandbox is a tool used for the dynamic analysis of malware.

Figure 3 shows the structure of Cuckoo Sandbox. It consists of four main parts; Cuckoo Host, Virtual Network, Cuckoo Guest and the Internet. The Cuckoo host is mainly responsible for the guest operations management, start and finish the analysis process, dispose network traffic and generate behavioral reports. The Cuckoo guest (also known as the Analysis Guest) is the environment that connected to the Cuckoo host through a virtual network (or a switch). The main task for the Cuckoo Guest is to execute a malware sample to analyze its files and then generate a report to send it back to the Cuckoo host through the same virtual switch. The system can have one or many Cuckoo guests, each of them run and execute unique malware sample. This provides efficiency to the system performance as well as increase the speed the process. In case of malware samples that need an internet connection to be executed, the Cuckoo host uses the internet network for this purpose by passing the traffics that were produced by the Cuckoo guest.

The environment used for this research was set on a Mac OS. Since the dataset requires a Windows OS, a version of Windows 10 was installed on VirtualBox ( which is a cross-platform virtualization software that allow an OS to run as a software program on another type of OS). To execute and analyze malware samples accurately, the virtual Windows 10 must be a clean environment for running and testing, that’s why all applications and any configurations that would make noisy network traffics were disabled.

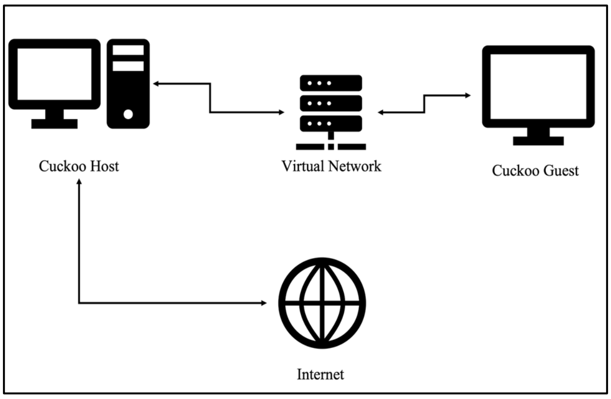

Fig. 3 The Structure of Cuckoo Sandbox Environment

## 3.3. Deep Learning Models

To build a prediction model using deep learning, the data must be split into two portions, one for training the model to predict an outcome which is called a Training set and another one for validating the accuracy of the model which is called Testing set. Such way is known as train-test split [28]. However, splitting the dataset might result in an unacceptable situation such as overfitting; which is a state occurs when the model is trained too well that it became sensitive to the details and the noise of the data. Consequently, this can be avoided using a cross-validation method, which is a powerful way to train and test a model by splitting the dataset into fixed number of folds (10 by default). The model uses 10 folds – (9 training, 1testing). The process is repeated x10 times by changing the testing fold each time [28]. Thus, this research used cross-validation to test the applied models.

By using cross-validation, many deep learning models predict the behavior of crypto-mining malware. Nevertheless, as it can be seen from the literature review, there is no specific method that works well in all different cases. For that, this research applied three different methods to provide the best prediction results. These are; Long Short-term Memory (LSTM), Support Vector Machine (SVM) and Random Forest (RF).

### 3.3.1. LSTM

Long short-term memory (LSTM) is a type of recurrent neural network (RNN) that used mainly in deep learning studies. The purposes for using LSTM are classification, processing and prediction of future events depend on data of time series. The LSTM deals with time data and it is well-suited for this type of data because it overcame the problem of unknown lags of time durations that used to show in normal RNN methods [31].

### 3.3.2. SVM

Support vector machines (SVMs) are type of learning method that well-known for dealing with data regression, classification and the detection of outliers. The SVM predicts the points by first classifying it as positive or negative and place it on the hyperplane based on the classes being predicted. SVM is a Supervised Machine Learning Algorithm with the focus on the data regression [32].

### 3.3.3. RF

The Random forest (RF) model is machine learning classifier that focus on the ability to predict an output based on multiple combined regression decision trees. Each of which is built separated from one another by using a random vector that was tagged from the input sample. Same as SVM, Random forest is a Supervised method. However, not only focus on the data regression but also on the data classification problems [34].

All the three deep learning models are exposed to the prepared datasets to be trained and tested. These models serve different purposes and hence can show more accurate results.

## 4. Results and Discussion

### 4.1. Evaluation metrices of The Models

The performance of the deep learning classifiers can be measured using the confusion matrix; also known as error matrix [29]. It consists of four terms:

1. True positive (TP) rate; refers to the numbers of records that were predicted correctly as positive.

2. False positive (FP) rate; is the opposite of TP, which indicates the incorrectly predicted records as positive.

3. True negative (TN) rate; refers to the numbers of records that were predicted correctly as negative.

4. False negative (FN) rate; is the opposite of TN, which indicates the incorrectly predicted records as negative.

These terms are used as input variables to measure four performance characteristics, including Accuracy, Precision, Recall and F1-Measure. Their description and measurements are as the following:

1. Accuracy: refers to the effectiveness of the classifier model, its equation as follows (1):

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

2. Precision: represents the power of prediction for the model, its equation as follows (2):

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

3. Recall: indicates how sensitive the model is, its equation as follows (3):

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

5. F1-Measure: measures the balance between the Precision and the Recall of a model, its equation as follows (4):

$$F1 - Measure = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)} \tag{4}$$

After pre-processing of the dataset, both extracted opcodes from IDA Pro and generated reports of system calls from Cuckoo Sand box were trained and tested by the deep learning models.

### 5.1. Static Analysis of Deep Learning Models On Opcodes

After applying the three mentioned deep learning methods using WEKA tool on the opcodes dataset which is the combination of Op_ benign and Op_crypto files from table 2, the data were modeled using cross-validation with 10 folds and 10 repetitions. Figure 4 below illustrates the findings of the performance of the three classifiers. The chart showed that all three classifiers achieved good scores and above 95 %. However, LTSM outperformed both RF and SVM in all four metrices; accuracy (98%), precision (97%), recall (97%) and F-1 (97%). Moreover, both SVM and RF scored almost similar results. Table 4 shows the percentage of these classifiers.

Table. 4 The Performance of Classifiers using Opcodes Dataset

| Classifier | Accuracy | Precision | Recall | F1-Measure |
|---|---|---|---|---|
| LSTM | 98% | 97% | 97% | 97% |
| SVM | 95% | 96% | 95% | 94% |
| RF | 93% | 94% | 95% | 95% |

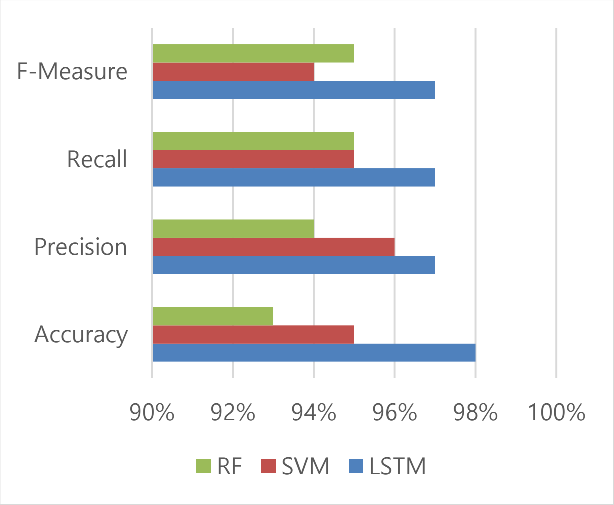

Fig. 4 Comparison of The Performance Between The Applied Deep Learning Methods of Opcodes Dataset

### 5.2. Dynamic Analysis of Deep Learning Models On System Calls

The same process that were applied on opcode dataset were also applied on system calls dataset. The results of classifiers performance are shown in figure 5. As noted from the chart, the performance of LTSM was somewhat better than the SVM and RF in terms of accuracy and precision. This due to that LTSM has an internal memory that can store the previous input. Table 5 summarizes the performance of these classifiers.

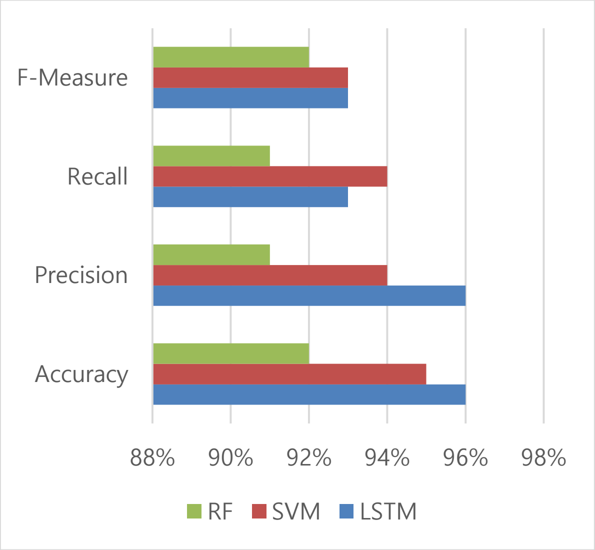

Fig. 5 Comparison of The Performance Between The Applied Deep Learning Methods of System Calls Dataset

Table. 5 The Performance of Classifiers using System Calls Dataset

| Classifier | Accuracy | Precision | Recall | F1-Measure |
|---|---|---|---|---|
| LSTM | 96% | 96% | 93% | 93% |
| SVM | 95% | 94% | 94% | 93% |
| RF | 92% | 91% | 91% | 92% |

From the results of both static and dynamic analysis, it can be concluded that both approached can detect a potential crypto-mining malware in a host-based environment. However, the static analysis achieved higher scores than the dynamic analysis but that should not always mean that it is more accurate. The dynamic analysis revolves around detecting the crypto-malware while running the code which captures a real event. The performance of the classifiers might be lower comparing with the performance of static analysis but it is more accurate and more realistic.

## 6. Conclusion and Future Work

The present and future of the financial world revolves around cryptocurrency and its applications. With its popularity being increased constantly, many breaches are discovered that threat the safety and security of these applications. One of the most dangerous threats is the crypto-jacking that an exploiter uses a victim's hardware to generate illegally cryptocurrency. Hence, the aim of this research was to make use of deep learning methods in predicting the potential crypto-mining malware in a host-based environment. Three well-known classifiers were used. The research was conducted based on an two datasets; Opcodes files that were extracted from binary sequential code using IDA Pro tool, and System calls files that were processed using the Cuckoo Sandbox tool. After preprocessing the data, the three models were built and trained to detect crypto-malware actions and behaviors. The findings compared the mentioned deep learning methods based on their rate of accuracy, precision, recall and f1-Measure. It can be concluded that LTSM outperformed both SVM and RF in both static and dynamic analysis and was able to reach a performance rate of 98% and 96% in static and dynamic approaches, respectively. Future studies should be conducted on a larger set of data with more attributes to have better and more accurate prediction results.

## References

[1] Darabian, Hamid & Homayounoot, Sajad & Dehghantanha, Ali & Hashemi, Sattar & Karimipour, Hadis & Parizi, Reza & Choo, Kim-Kwang Raymond. (2020). Detecting Cryptomining Malware: a Deep Learning Approach for Static and Dynamic Analysis. Journal of Grid Computing. 18. 10.1007/s10723-020-09510-6.

[2] A. Pastor et al., "Detection of Encrypted Cryptomining Malware Connections With Machine and Deep Learning," in IEEE Access, vol. 8, pp. 158036-158055, 2020, doi: 10.1109/ACCESS.2020.3019658.

[3] A. Marshall, "Combined crypto market capitalization races past $800 bln," https://cointelegraph.com/news/combined-crypto-mar ket- capitalization- races- past- 800- bln, accessed: 2020-02-28.

[4] Hemdan, E.ED., El-Shafai, W. & Sayed, A. CR19: a framework for preliminary detection of COVID-19 in cough audio signals using machine learning algorithms for automated medical diagnosis applications. J Ambient Intell Human Comput (2022). https://doi-org.sdl.idm.oclc.org/10.1007/s12652-022-03732-0

[5] J. Park, S. Park, K. Kim and D. Lee, "CORUS: Blockchain-Based Trustworthy Evaluation System for Efficacy of Healthcare Remedies," 2018 IEEE International Conference on Cloud Computing Technology and Science

[6] Dimiduk, D.M., Holm, E.A. & Niezgoda, S.R. Perspectives on the Impact of Machine Learning, Deep Learning, and Artificial Intelligence on Materials, Processes, and Structures Engineering. Integr Mater Manuf Innov 7, 157–172 (2018). https://doi.org/10.1007/s40192-018-0117-8

[7] Sinnott, Richard & Wu, Fang & Chen, Wenbin. (2018). A Mobile Application for Dog Breed Detection and Recognition Based on Deep Learning. 87-96. 10.1109/BDCAT.2018.00019.

[8] F. Z. Meskini and R. Aboulaich, "Multi-agent based simulation of a smart insurance using Blockchain technology," 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS), 2019, pp. 1-6, doi: 10.1109/ICDS47004.2019.8942270.

[9] M. A. Razali and S. M. Shariff, "Cmblock: In-browser detection and prevention cryptojacking tool using blacklist and behavior- based detection method," in International Visual Informatics Con- ference (IVIC). Springer, 2019, pp. 404–414.

[10] A. D. Yulianto, P. Sukarno, A. A. Warrdana, and M. Al Makky, "Mitigation of cryptojacking attacks using taint analysis," in 2019 4th International Conference on Information Technology, Infor- mation Systems and Electrical Engineering (ICITISEE). IEEE, 2019, pp. 234–238.

[11] M. Caprolu, S. Raponi, G. Oligeri, and R. Di Pietro, "Crypto mining makes noise," arXiv:1910.09272, 2019.

[12] J. Liu, Z. Zhao, X. Cui, Z. Wang, and Q. Liu, "A novel approach for detecting browser-based silent miner," in 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC). IEEE, 2018, pp. 490–497.

[13] J. Rauchberger, S. Schrittwieser, T. Dam, R. Luh, D. Buhov, G. Po ̈tzelsberger, and H. Kim, "The other side of the coin: A framework for detecting and analyzing web-based cryptocurrency mining campaigns," in Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES), 2018, pp. 1–10.

[14] I. Petrov, L. Invernizzi, and E. Bursztein, "Coinpolice: De-tecting hidden cryptojacking attacks with neural networks," arXiv:2006.10861, 2020.Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond (2010)

[15] Hassan, Nurul & Jain, Nishchay & Chandna, Vinay. (2018). BLOCKCHAIN, CRYPTOCURRENCY AND BITCOIN.

[16] Laila, Fetjah & Azbeg, Kebira & Ouchetto, Ouaïl & jai andaloussi, Said. (2021). Towards a Smart Healthcare System: An Architecture Based on IoT, Blockchain, and Fog Computing. International Journal of Healthcare Information Systems and Informatics. 16. 1-18. 10.4018/IJHISI.20211001.oa16.

[17] Ullah, I. et al. (2022) 'Certificate-Based Signature Scheme for Industrial Internet of Things Using Hyperelliptic Curve Cryptography', Wireless Communications & Mobile Computing, pp. 1–8. doi: 10.1155/2022/7336279.

[18] Dar, MA, Askar, A, Alyahya, D & Bhat, SA 2021, 'Lightweight and Secure Elliptical Curve Cryptography (ECC) Key Exchange for Mobile Phones', International Journal of Interactive Mobile Technologies, vol. 15, no. 23, pp. 89–103.

[19] Bitcoin.org. 2022. Bitcoin - Open source P2P money. [online] Available at: <https://bitcoin.org/en/> [Accessed 12 March 2022].

[20] Alkaeed, MK, Alamro, Z, Al-Ali, MS, Al-Mohammed, HA & Khan, KM 2020, 'Highlight on Cryptocurrencies Mining with CPUs and GPUs and their Benefits Based on their

Characteristics', 2020 IEEE 10th International Conference on System Engineering and Technology (ICSET), System Engineering and Technology (ICSET), 2020 IEEE 10th International Conference on, pp. 67–72.

[21] Wheeler, KA & Bowers, AW 2019, 'A Comparative Power Quality Analysis of Cryptocurrency Mining Loads', 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Electrical and Computer Engineering (CCECE), 2019 IEEE Canadian Conference of, pp. 1–5

[22] Nadeau, M., 2022. What is cryptojacking? How to prevent, detect, and recover from it. [online] CSO Online. Available at: <https://www.csoonline.com/article/3253572/what-is-cryptojacking-how-to-prevent-detect-and-recover-from-it.html> [Accessed 12 March 2022].

[23] Ning, R, Wang, C, Xin, C, Li, J, Zhu, L & Wu, H n.d., 'CapJack: Capture In-Browser Crypto-jacking by Deep Capsule Network through Behavioral Analysis', Proceedings - IEEE INFOCOM, vol. 2019–April, pp. 1873–1881.

[24] E. Tekiner, A. Acar, A. S. Uluagac, E. Kirda and A. A. Selcuk, "SoK: Cryptojacking Malware," 2021 IEEE European Symposium on Security and Privacy (EuroS&P), 2021, pp. 120-139, doi: 10.1109/EuroSP51992.2021.00019.

[25] Zimba, A, Zhaoshun Wang, Hongsong Chen & Mulenga, M 2019, 'Recent Advances in Cryptovirology: State-of-the-Art Crypto Mining and Crypto Ransomware Attacks', KSII Transactions on Internet & Information Systems, vol. 13, no. 6, pp. 3258–3279.

[26] J. Ru ̈th, T. Zimmermann, K. Wolsing, and O. Hohlfeld, "Digging into browser-based crypto mining," in Proceedings of the Internet Measurement Conference (IMC) 2018, 2018, pp. 70–76.

[27] "Browser-based deep behavioral detection of web cryptomining with coinspy," in Workshop on Measurements, Attacks, and De-fenses for the Web (MADWeb) 2020, 2020, pp. 1–12.

[28] H. N. C. Neto, M. A. Lopez, N. C. Fernandes, and D. M. Mattos, "Minecap: super incremental learning for detecting and blocking cryptocurrency mining on software-defined networking," Annals of Telecommunications, pp. 1–11, 2020.

[29] W.Wang, B.Ferrell, X.Xu, K.W.Hamlen, and S.Hao,"Seismic: Secure in-lined script monitors for interrupting cryptojacks," in European Symposium on Research in Computer Security (ES- ORICS). Springer, 2018, pp. 122–142.

[30] R.K.Konoth, E.Vineti, V.Moonsamy, M.Lindorfer, C.Kruegel, H. Bos, and G. Vigna, "Minesweeper: An in-depth look into drive- by cryptocurrency mining and its defense," in Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS), 2018, pp. 1714–1730.

[31] "Browser-based deep behavioral detection of web cryptomining with coinspy," in Workshop on Measurements, Attacks, and De- fenses for the Web (MADWeb) 2020, 2020, pp. 1–12.

[32] M. Musch, C. Wressnegger, M. Johns, and K. Rieck, "Thieves in the browser: Web-based cryptojacking in the wild," in Pro- ceedings of the 14th International Conference on Availability, Reliability and Security (ARES), 2019, pp. 1–10.

[33] W. Bian, W. Meng, and M. Zhang, "Minethrottle: Defending against wasm in-browser cryptojacking," in Proceedings of The Web Conference (WWW) 2020, 2020, pp. 3112–3118.

[34] I. Petrov, L. Invernizzi, and E. Bursztein, "Coinpolice: Detecting hidden cryptojacking attacks with neural networks," arXiv:2006.10861, 2020.

[35] M. Caprolu, S. Raponi, G. Oligeri, and R. Di Pietro, "Crypto mining makes noise," arXiv:1910.09272, 2019.

[36] J. Z. i Mun ̃oz, J. Sua ́rez-Varela, and P. Barlet-Ros, "Detecting cryptocurrency miners with netflow/ipfix network measurements," in 2019 IEEE International Symposium on Measurements & Networking (M&N). IEEE, 2019, pp. 1–6.

[37] H. Darabian, S. Homayounoot, A. Dehghantanha, S. Hashemi, H. Karimipour, R. M. Parizi, and K.-K. R. Choo, "Detecting cryptomining malware: a deep learning approach for static and dynamic analysis," Journal of Grid Computing, pp. 1-11, 2020.

[38] Yazdinejad, A., HaddadPajouh, H., Dehghantanha, A., Parizi, R., Srivastava, G., & Chen, M. (2020). Cryptocurrency malware hunting: A deep Recurrent Neural Network approach. Applied Soft Computing, 96, 106630. doi: 10.1016/j.asoc.2020.106630