Identifying Malware on Cyber Physical Systems by incorporating Semi-Supervised Approach and Deep Learning

IOP Conf. Series: Earth and Environmental Science **322** (2019) 012012 doi:10.1088/1755-1315/322/1/012012

Identifying Malware on Cyber Physical Systems by incorporating Semi-Supervised Approach and Deep Learning

Shaila Sharmeen, Shamsul Huda and Jemal Abawajy

School of IT, Deakin University, Melbourne, Australia
E-mail: ssharmee@deakin.edu.au, shamsul.huda@deakin.edu.au, jemal.abawajy@deakin.edu.au

Abstract. Malicious applications can be a security threat to Cyber-physical systems as the Cyber-physical systems are composed of heterogeneous distributed systems and mostly depends on the internet, ICT services and products. The usage of ICT products and the services gives the opportunity of less expensive data collection, intelligent control and decision systems using automated data mining tools. Cyber-physical systems become exposed to the internet and the public networks as it has integrated to the ICT networks for easy automated options. Cyber-attacks can lead functional failure, blackouts, energy theft, data theft etc. and this will be critical security concern of Cyber-physical systems. At present, the mobile devices are replacing the pc environment and become a key element of Internet of Things. Therefore, it is essential to develop such a malware detection engine that will identify the mobile malware and reduce the spreading of the malicious code through mobile devices. This research work will identify the malware by incorporating semi-supervised approach and deep learning. The original and significant contributions are to propose an effective malware detection model by incorporating semi-supervised approach and deep learning, to implement the model using parallel processing and to evaluate the performance of the model using recent dataset. Here we have used the permission and the API call as the features. The proposed method has been tested on the real mobile malware data set and it shows improvement in accuracy. The Experimental results show that the deep learning along with semi-supervised method will be an effective way to identify the malware and it outperforms other detection methods.

1. Introduction

Cyber-physical systems(CPS) are used in a large scale in the modern industrial system. Internet of Things shows a rapid growth, and this is why the usage of CPS has been also increased [1]. CPS integrates the physical systems with the computational process, software, services and able to do easy and efficient data processing over internet [2]. CPS allows the communication between humans, machines and products. IoT applications are immensely used in the CPS to increase the computational and storage capabilities as well as it will help to monitor, control and do the decision making for the remote devices. IoT applications help to maximize the producibility and throughput by enabling the efficient data collection, aggregation of data and make them available for business organizations [1]. As the CPS has direct impact on the producibility and revenue, malicious activities have been introduced to the cyber-physical system and it shows an exponential growth. Cyber-attacks by developing malicious code or malware has become a threat to the CPS as it can cause national security concerns, financial loss, malfunctioning of process and even complete shutdown of industrial and system operations.

The conventional power grid system has been emerged and evolved into Smart grid by incorporating the services and the products of the Information technologies. Different service packages of ICT have been introduced to operate, control and monitor the power system and this makes the traditional power system more reliable [3]. The delivery, controlling and monitoring system of smart grid have shown a satisfactory performance level while using these ICT products [8]. The major components of the smart grid are divided into two categories: system components and the network components. In system components, household appliances, renewable energy resources, smart meters, electric utility centre and service providers are included. Home Area network (HAN) and Wide Area Network(WAN) are used as the way to communicate to the smart grid. Home Area Networks connects the smart devices such as smart phone, tv, air conditioner etc. with the smart meter [3]. On contrast, the WAN has connected the smart meters, service providers and electric utility. The smart meter does the activity of a gateway and connects in-house smart devices to the external components. In the industrial area and business offices, the industrial area networks (IAN) and business area networks (BAN) are used respectively. Smart devices are also connected to the smart meter in IAN and BAN. Since the communications are done over heterogenous networks, then the security of the equipment, data and user is a great concern [8]. The smart gird can suffer from Denial of service attack, malware spreading, compromising communication equipment, replay attack, eavesdropping and traffic analysis etc [3]. Smart grid, a CPS has needed proper attention to ensure the security and maintain the productivity [7].

Traditional IT security and the cyber-attacks on CPS are different in many aspects [4]. The upgrade of applications needs more time than the traditional IT. Moreover, in some cases, we do not pay attention to the security measures in CPS [3]. Operation and controlling become the main objective here. As a result, malware developers can easily introduce malware which will harm the system. Malware protection and detection system is a crucial need to ensure the security of CPS. The malware can spread over the network and make the system a victim of it.

In the smart industry arena, smartphones are extensively integrated with industrial Internet of Things (IoT) networks [9]. As the CPS deploys heterogeneous IoT devices to meet a wide range of user requirements, smartphones are widely used as a media for controlling and communicating the IoT devices. As smartphones has replaces the personal computers, they have become a key component of CPS. However, the integration of mobile devices with industrial IoT networks (such as CPS) exposes the IoT devices to significant malware threats. Mobile malware is the highest threat to the security of IoT data, user's personal information, identity, and corporate/financial information. This paper analyzes the efforts regarding malware threats aimed at the devices deployed in industrial mobile-IoT networks and related detection techniques.

The features of the smartphone such as web browsing, navigation, sending emails, share contents etc. make it an alternative of the personal computers. In this current era, people use the smart phone more frequently than the laptops or personal computers. Moreover, the high-speed RAM and processor make the smart phone an alternative choice of the personal computer. Making calls is not the only purpose of using smart phones, people frequently download and uses the applications that are useful in daily life such as making appointments, paying bills, banking activities, email, video conferencing, knowing the weather, news, shopping and much more. To improve the productivity and interoperability, business owners and service providers has migrated their products and service to the smartphone platform. As a result, the smart phone has become a growing trend to the business owners as well as to the users [10]. Android is the most popular and mostly used smart phone operating system which has occupied the 87% market share in 2018 [5,6,14]. More than 3millions applications has been introduced in the Android Official Apps store, Google play and nearly 65 billion of downloads of these applications has been occurred [15]. In addition, unverified applications available in third party apps stores. The flexible nature of Android market place and immense popularity of Android have made it a lucrative target for the malicious apps developers [10]. Cybercriminals have been focused on the Android platform and introduced malicious applications in the Android market place [13]. Malware shows an exponential growth in Android platform and in every 10 seconds, a malicious application has been released [15].

More than 9 million of malware have been found globally so far. Since Android has become the ultimate target of the malware developers, a proper malware detection engine is essential. An infected smart

phone device can capture the data, information, the controlling of the other IoT devices and will be an ultimate threat to the cyber-physical system. A proper malware detection mechanism will not only provide the security to the smart phone but also ensure a secure environment of CPS. Thus, we can minimize the security threat to the CPS in a great extent.

The main contribution of this paper is to design and implement a proper and efficient malware detection engine using semi-supervised approach and deep learning, that will detect the mobile malware and will make a safe environment for the CPS. We have analysed the performance of our model using the most recent real-life data set and compared with the recent works in this domain. Parallel processing has also been introduced to minimize the computational time. Experiment results show that our model outperforms over existing malware detection engine.

The rest of the paper is organised as follows: the problem is defined in the section 2. Section 3 describes the components of malware detection system. Deep learning and semi-supervised method are discussed in section 4. Section 5 represents our model. Experimental result and comparison are depicted in section 6. Section 7 describes the limitations and scope of our model. Conclusion and future work are discussed in section 8.

2. Problem Definition

To identify the unknown malwares and the benign apps into two different classes is our research goal. As Binary classification involves only two set of possible classes, we can represent this problem as a binary classification problem. We can consider all apps as a set of applications, A. There are different kinds of applications are available in A.

$$A = \{A_1, A_2, A_3, \ldots\ldots, A_n\} \quad (1)$$

In this A set there are n number of applications. These applications may be malware, benign ware and new unknown applications. Each application A_i has been defined with some feature vector F_i and class label CL_i.

$$A_i = \{F_i, CL_i\} \quad (2)$$

Here, F_i is the feature vector where k is the number of selected features.

$$Fi = \{F1, F2, F3, \ldots\ldots, Fk\} \quad (3)$$

Class label CL_i defines the label of the class where there is a label for benign ware, malware, and Unknown apps.

$$CL_i = \{CL_b, CL_m, CL_{uap}\} \quad (4)$$

Here, CL_b is the class label of benign ware, CL_m is the class label of malware and CL_{uap} is the class label of unknown apps.

We need to introduce a detection engine D that will classify the CL_{uap} into the two classes CL_b or CL_m. The detection engine will have a malware detection function (μ) and the achieved accuracy level (Acc_l). The detection engine will use the detection mechanism to classify the set of apps A by providing each of them a label of CL_b or CL_m. The definition of the Malware detection function is given below:

$$\mu(F): F \rightarrow CL \quad (5)$$

3. Components of Malware Detection System

The malware detection engine identifies the malware and benign applications. It consists of a collection of data, a feature extractor, feature selector and the classifier. In the detection engine, the data set refers malwares and the benign wares from different online market places. The feature extractor extracts a huge number of static, dynamic and hybrid features. The feature selectors select some of these features which are more relevant and necessary than other features. If the relevant features are selected, then this will provide a positive impact on the achieved accuracy level of the detection engine. The classifier will classify the applications into two groups: malware and benign ware. Different classifying algorithms

can be used in the detection engine such as Naïve based, random forest, j48, random tree etc. Figure 1 represents the malware detection engine.

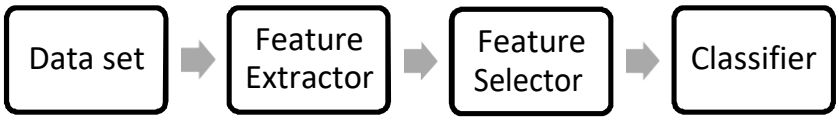

Figure 1. Malware Detection Engine

4. Integration of Semi-Supervised learning and Deep Learning

The high performance of Deep learning primarily depends on large volume of labelled data set. Collecting a large number of malware dataset is a big challenge but in contrast downloading unlabelled android applications from various market place is quite easier. That is why to use the knowledge from these easily available unlabelled android applications we choose the semi supervised approach with deep learning. The semi-supervised approach is the combination of supervised and unsupervised learning process. The semi supervised approach provides some specific benefits. Such as, it makes the detection process more accurate, effective and faster than the supervised method. The use of labelled and unlabelled data facilities the detection process. In supervised learning, only the labelled data is used [12,17]. At the very beginning, the features are extracted and selected from the labelled data. The data sets are divided in two categories: train data set and the test data set. The classifier is trained depending on the selected features of the train data set. After training the model, test data is applied to the classifier. The accuracy of the detection is measured using the comparison of actual labelling of test data and the determined labelling by the classifier. The supervised method suffers from specific limitations as the labelling of the data is quite expensive and time consuming [17]. In unsupervised learning, the process of clustering does the classification in such a way that the similarities within the cluster is maximized whereas the similarities among groups is minimized. The unlabelled data can be used in unsupervised learning. Unsupervised classification put the same type of data in a group and the patterns within the group have high similarities [12]. The similarities within the group member is high and the dissimilarity with the members of another group is also high. Minkowski distance, Maximum Likelihood, expectation maximization etc. are used to extract the intrinsic patterns. As malware classification is a binary classification, so the unsupervised classification cannot be applied here [12]. However, the intrinsic behaviour from unsupervised learning can play a vital role in supervised malware detection process. The semi supervised method incorporates the measured distance feature to the labelled data set of supervised learning.

Deep learning is a new area of machine learning research that plateau at a certain level of accuracy when the network is feed with large number of examples. Deep learning algorithms are improved at great level in recent years and showed exceptional accuracy than ever before. This higher level of accuracy makes it an obvious choice in various sectors like automated driving, aerospace, defence system, medical research to advertisement industry. The key advantage of deep learning is that it can continue to improve its accuracy as the size of label data increases. As a relatively new concept, a lot of new libraries are under development and currently available. By all measures TensorFlow, was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization, is the undisputed leader [26].

5. Proposed Malware Detection Model

Our proposed model is a composite model for detecting the Android malware where we have combined the semi-supervised approach with deep learning. This approach will be able to accumulate the strength of both supervised and unsupervised learning to produce a single classifying mechanism. We have constructed our data set by collecting the known malware, known benign ware and the unknown applications. Here the known malware and known benign ware will be served as the labelled data whereas the unknown applications will be considered as unlabelled data. Using the unlabelled data, we do the classification of the applications which is known as unsupervised classification. We train the deep

learning model incorporating the knowledge from both labelled and unlabelled data. After completion of the training process, we provide sample applications to test the detection engine. As the training process has incorporate the knowledge of both labelled and unlabelled data, the accuracy of the detection engine will be enhanced. Moreover, in our model, we have used the deep learning framework TensorFlow to train and test the data. GPU based parallel processing has been applied here to reduce the computational cost.

5.1. Data Collection

We collected 8119 malware samples from different families. Collected 5560 malwares from Drebin data set [21]. Other malwares are collected from Androzoo [22]. 10000 benign wares are collected from Drebin data set [21]. 10000 unlabelled applications are collected from Androzoo [22]. 1957 applications, downloaded from the ApkPure [23] and ApkMirror [24] serves here as the unlabelled data. The following table will describe the source, number and nature of the applications.

Table 1. Data Set Details

Nature	**Number**	**Source**	**Total**
Malware	5560	Drebin data set [21]	8119
	2559	Androzoo [22]	
Benign	10000	Drebin data set [21]	10000
Labelled Applications		Malware and benign ware	18119
Unlabelled Applications	10000	Androzoo [22]	11957
	1957	ApkPure and ApkMirror [23,24]	

5.2. Feature extraction and Feature Selection

In order to minimize the computational cost, we have considered the static features only. We have extracted the permission and the API call features from both labelled and unlabelled data set. A python script extracts the API call and the permissions using the functions of Androguard [25] and generates a global set. We have selected 502 API call and permissions using information gain selection method.

5.3. Proposed Malware Detection Algorithm

We have introduced the malware detection by incorporating semi supervised technique with Deep learning. k-means clustering has been considered as unsupervised learning method. We have measured the Euclidean distance of the instance from the cluster centres. We have accommodated this distance feature in the feature vector. Thus, we have integrated the unsupervised learning knowledge to the supervised learning. We trained the model using the deep learning framework named TensorFlow [16] and then executed on our test dataset to evaluate the performance of our model.

Algorithm: Malware Detection using Semi-Supervised Method and Deep learning

1. input ← DL(F_1F_2, F_3 ... ... F_m) {Labeled data set constructed from known malware and benign applications with m number of selected features}
2. input ← UDL(F_1F_2, F_3 ... ... F_m) {Unlabeled data set with m number of selected features}
3. Output Acc_1, Acc_2.
4. Begin
5. Split the labeled data into train and test data set. Train and test the model for DL using TensorFlow and Calculate accuracy, Acc_1. {supervised method using deep learning}
6. Remove the level of the data DL and Construct a new data set UDL_1 by doing union of
   DL and UDL

7. Find the clusters using K-means algorithm using UDL_1. {unsupervised method using selected features set}
8. Find the cluster centers.
9. Compute distance of instance from the cluster center using Euclidean distance.
10. Inject the distance in the feature vector DL and form a new feature vector DL_1. {combine the unsupervised learning to supervised learning}
11. Split the labeled data into train and test data set. Re-Train and test the model for DL_1using TensorFlow and Calculate accuracy, Acc_2. {semi supervised method using deep learning}
12. END

6. Experimental Results and Comparison

We have implemented the algorithm in python and evaluated the performance of our proposed model. Known malware and benign ware are served as labelled data in the algorithm. We have measured the accuracy, recall, ROC_AUC_Score and f1_score as the performance metrics. Accuracy is the percentage of correctly identified apps. Accuracy can be defined using the following equation,

$$\mathrm{ACC} = (\mathrm{TP} + \mathrm{TN})/(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}) \quad (6)$$

Recall is the number of correct results divided by the number of results that should have been returned.

$$\mathrm{Recall} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \quad (7)$$

Accuracy alone cannot define the model. ROC_AUC Curve defines capability of distinguishing between classes. Higher ROC_AUC curve score is more desirable. F-measure or f1_score illustrate the balance between recall and precision and can be calculated using the following equation.

$$\mathrm{F1} - \mathrm{score} = 2 * (\mathrm{Precision} * \mathrm{Recall})/(\mathrm{Precision} + \mathrm{Recall}) \quad (8)$$

Our proposed model has been achieved a higher accuracy level and f1_score than supervised deep learning method. Semi-supervised model with deep learning has shown better performance than the deep learning model. Table 2 represents the experiment results.

Table 2. Experiment Result

Method	**Accuracy**	**Recall**	**ROC_AUC_SCORE**	**f1_score**
Deep Learning	0.9594	0.9617	0.9639	0.9606
Semi-supervised Deep Learning	0.9715	0.9541	0.9650	0.9627

We have compared the performance of our model with other approaches which is represented in the Table 3. Our model showed better performance than other detection engines. Some research works [14,20] provide better accuracy level than our model as they have not considered recent malware set.

Table 3. Comparison with other Research works

Ref.	**Accuracy**
[11]	97%
[12]	91.98%
[15]	93.62%
[18]	96.69%
[19]	96.9%
[21]	93.9%
Semi-supervised Deep Learning	97.15%

Although from the comparison table we can notice that some other machine learning classifiers show nearly equal level of accuracy, the key advantage of our proposed deep learning model is that the accuracy will keep improving from adding both labelled and unlabelled android applications. Moreover, in our experiment we picked GPU supported TensorFlow library to minimize the computational time. To get the benefit of parallel processing we used system having 8 GPUs.

7. Limitation and Scope

Involving only the static feature is one of the main limitations of our model. The static features will not be able to detect the runtime attacks like update attack, mimicry attack etc. In out experiment, we have considered only the static features to minimize the computational cost as the mobile platform suffers from the shortage of available resources. In future, we will incorporate the dynamic features in our model and will be able to support zero-day detection.

8. Conclusion and Future Work

In this paper, we have proposed and implemented a malware detection model for mobile-IoT applications in CPS. We have integrated the semi-supervised approach with the deep learning. This model has combined the benefits of semi-supervised model and deep learning which become the key strength of our model. We overcome the limitations of supervised learning and incorporated the knowledge from unsupervised learning. This incorporation of knowledge has enhanced the opportunity to identify the new malware. Moreover, we have considered the API call along with the permission feature to increase the accuracy of our proposed model. As the permission feature alone cannot illustrate the behaviour of the applications, we included the API call features as well. This helped the detection engine to generate the more accurate signature of the application. Since malware has adapted advanced techniques to hide the malicious nature and make variants frequently, we collected the most recent malware to depict the real current scenario and evaluate the performance of our model using the recent malware data set. In addition, we have introduced parallel processing to reduce the computational time.

Although, our proposed model has provided satisfactory performance, still there is a future scope to improve the accuracy level by involving the dynamic features. In future, we will train and test our model using both static and dynamic features. Moreover, we will consider most recent malware data set and test our model using both labelled and unlabelled data and will able to detect new and unknown malware. This proposed malware detection technique can be deployed to enhance the security of CPS.

9. References

[1] Evans, D., 2011. The internet of things: How the next evolution of the internet is changing everything. *CISCO white paper*, *1*(2011), pp.1-11.

[2] Monostori, L., 2014. Cyber-physical production systems: Roots, expectations and R&D challenges. *Procedia Cirp*, *17*, pp.9-13.

[3] Aloul, F., Al-Ali, A.R., Al-Dalky, R., Al-Mardini, M. and El-Hajj, W., 2012. Smart grid security: Threats, vulnerabilities and solutions. *International Journal of Smart Grid and Clean Energy*, *1*(1), pp.1-6.

[4] Cardenas, A., Amin, S., Sinopoli, B., Giani, A., Perrig, A. and Sastry, S., 2009, July. Challenges for securing cyber physical systems. In *Workshop on future directions in cyber-physical systems security* (Vol. 5).

[5] https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating systems/ [Access Date: 30.11.2018]

[6] Ping Yan and Zheng Yan, A survey on dynamic mobile malware detection, *Software Quality Journal*, September 2018, Volume 26, Issue 3, pp 891–919

[7] Sridhar, S., Hahn, A. and Govindarasu, M., 2012. Cyber-Physical System Security for the Electric Power Grid. *Proceedings of the IEEE*, *100*(1), pp.210-224.

[8] Mo, Y., Kim, T.H.J., Brancik, K., Dickinson, D., Lee, H., Perrig, A. and Sinopoli, B., 2012. Cyber–physical security of a smart grid infrastructure. *Proceedings of the IEEE*, *100*(1), pp.195-209.

[9] Sadeghi, A.R., Wachsmann, C. and Waidner, M., 2015, June. Security and privacy challenges in industrial internet of things. In *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE* (pp. 1-6). IEEE.
[10] Odusami M, Abayomi-Alli O, Misra S, Shobayo O, Damasevicius R, Maskeliunas R. Android Malware Detection: A Survey. In *International Conference on Applied Informatics 2018 Nov 1* (pp. 255-266). Springer, Cham.
[11] Li D, Wang Z, Xue Y. Fine-grained Android Malware Detection based on Deep Learning. In *IEEE Conference on Communications and Network Security (CNS)* 2018 May 30 (pp. 1 -2).
[12] Arora A, Peddoju SK, Chouhan V, Chaudhary A. Poster: Hybrid Android Malware Detection by Combining Supervised and Unsupervised Learning. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking 2018* Oct 15 (pp. 798-800). ACM.
[13] Aneja L, Babbar S. Research Trends in Malware Detection on Android Devices. In *International Conference on Recent Developments in Science, Engineering and Technology* 2017 Oct 13 (pp. 629-642). Springer, Singapore.
[14] Li D, Wang Z, Li L, Wang Z, Wang Y, Xue Y. FgDetector: Fine-Grained Android Malware Detection. *In Data Science in Cyberspace (DSC), 2017 IEEE Second International Conference on 2017* Jun 26 (pp. 311-318). IEEE.
[15] Li J, Sun L, Yan Q, Li Z, Srisa-an W, Ye H. Significant Permission Identification for Machine Learning Based Android Malware Detection. *IEEE Transactions on Industrial Informatics. 2018 Jan 12.*
[16] TensorFlow Framework available at https://www.tensorflow.org/ [Access Date: 30.11.2018]
[17] Kotsiantis SB, Zaharakis I, Pintelas P. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering. 2007* Jun 10;160:3-24.
[18] Chuang, H.Y. and Wang, S.D., 2015, August. Machine learning based hybrid behavior models for Android malware analysis. In *Software Quality, Reliability and Security (QRS), 2015 IEEE International Conference on* (pp. 201-206). IEEE.
[19] Dini G, Martinelli F, Saracino A, Sgandurra D. MADAM: a multi-level anomaly detector for android malware. In *International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security 2012* Oct 17 (pp. 240-253). Springer, Berlin, Heidelberg.
[20] Wu DJ, Mao CH, Wei TE, Lee HM, Wu KP. Droidmat: Android malware detection through manifest and api calls tracing. In *Information Security (Asia JCIS), 2012 Seventh Asia Joint Conference on 2012* Aug 9 (pp. 62-69). IEEE.
[21] Arp D, Spreitzenbarth M, Hubner M, Gascon H, Rieck K, Siemens CE. DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket. In Ndss 2014 Feb *23* (Vol. 14, pp. 23-26).
[22] Allix K, Bissyandé TF, Klein J, Le Traon Y. Androzoo: Collecting millions of android apps for the research community. In *Mining Software Repositories (MSR), 2016 IEEE/ACM 13th Working Conference on 2016 May 14* (pp. 468-471). IEEE.
[23] https://apkpure.com/
[24] https://www.apkmirror.com/
[25] https://github.com/androguard/androguard
[26] https://www.kdnuggets.com/2018/04/top-16-open-source-deep-learning-libraries.html