Global Scientific JOURNALS

GSJ: Volume 10, Issue 2, February 2022, Online: ISSN 2320-9186
www.globalscientificjournal.com

SECURITY CONCEPT IN INTERNET OF THINGS USING SUPERVISED & UNSUPERVISED MACHINE LEARNING ALGORITHM

Adeleye, S. A[1]., Obruche C. O[2]. Idama R. O[3]. Iyamah B. E[4]., Oladele K. J[5].Aworonye E. A[6]

Department of Computer Science

Faculty of Science

Federal University of Petroleum Resources (FUPRE)

Abstract

An Internet of Things (IoT) architecture generally consists of a wide range of Internet-connected devices or things such as Android devices, and devices that have more computational capabilities (e.g., storage capacities) are likely to be targeted by ransomware authors. In this work, we present a machine learning based approach to detect ransomware attacks by monitoring power consumption of Android devices. Specifically, our proposed method monitors the energy consumption patterns of different processes to classify ransomware from non-malicious applications. We then demonstrate that our proposed approach outperforms K-Nearest Neighbors, Neural Networks, Support Vector Machine and Random Forest, in terms of accuracy rate, recall rate, precision rate and F-measure

KEYWORDS: IOT devices, Android device, Cyberattacks

1.0 Introduction

According to IBM, devices connected to the Internet are expected to exceed the number of human beings and the evolution of connectivity is expected to continue such that by 2020 the number of connected devices will be around 50 billion. This proliferation of connected devices in an actuating network has created what has become known as the Internet of Things (IoT). A platform in which sensors and actuators blend seamlessly with the environment to share information in order to develop a common operating picture (Gubbi et al., 2013).

Pervasive growth of Internet of Things (IoT) is visible across the globe. The 2016 Dyn cyberattack exposed the critical fault-lines among smart networks. Security of Internet of Things (IoT) has become a critical concern. The danger exposed by infested Internet-connected things not only affects the security of IoT, but also threatens the complete Internet eco-system which can possibly exploit the vulnerable Things (smart devices) deployed as botnets. Mirai malware compromised the video

surveillance devices and paralyzed Internet via distributed denial of service (DDoS) attacks. In the recent past, security attack vectors have evolved bothways, in terms of complexity and diversity. Hence, to identify and prevent or detect novel attacks, it is important to analyze techniques in IoT context. This survey classifies the IoT security threats and challenges for IoT networks by evaluating existing defense techniques.

An IoT system starts from the level where a single object is identified using a unique global identifier which is globally addressable. The level of information obtained by accessing the object, in this case, can be as low as static data that is stored on the radio frequency identification (RFID) tags. IoT is therefore described as objects with a unique identifier, having Internet connectivity; is (interactively) accessible by other objects herein referred to as the ‘‘things’’. IoT has stepped out of its infancy and is the next revolutionary technology in the transformation of Internet into a fully integrated future Internet (of things). This development is fuelled by the recent increase in adoption and integration of wireless network technologies, Wireless Sensor Networks (WSN), RFID tags, as well as actuating nodes. On this concept, Karimi and Atkinson claimed, expanding communication networks to include physical objects will further accelerate the number of connected devices, as well as the amount of information that can be shared through the Internet (Karimi and Atkinson, 2013).

IoT presents ubiquitous connectivity for a wide range of devices, services, and applications. These include intelligent computers, smart-phones, office equipment, wireless enabled cars, lighting systems, heating, and ventilation and air-condition (HVAC), household appliances, and many others. To be IoT-enabled, a device (‘thing’) ought to be on a network and connected to a communicating node. Various communication network technologies (infrastructures) such as 3G, LTE, Wi-Fi, Bluetooth, ZigBee, Z-wave, Sigfox, etc. provide connectivity services for IoT deployment on many services platforms.

2.0 Materials and Methods

Thing, 2017: analyzed IEEE 802.11 network threats and proposed an anomaly network IDS to detect and classify attacks in IEEE 802.11 networks. This work is considered as the first work that employ deep learning algorithms for IEEE802.11 standard. Thing experimented Stacked Auto-encoder (SAE) architecture with both two and three hidden layers.

The author experienced different activation functions for the hidden neurons. To test his strategy, he used a dataset generated from a lab emulated Small Office Home Office (SOHO) infrastructure. He achieved an overall accuracy of 98.66% in a 4-class classification (legitimate traffic, flooding type attacks, injection type attacks and impersonation attacks).

Diro et al., 2017: recommended the use of the fog computing in IoT systems to detect intrusions. Fog computing is about equipping the fog layer (hubs, routers or gateways) with an intelligent data processing at an intermediate level in the

aim to improve efficiency and reduce the data transported to the cloud. Such a technology enables distributed attack detection which is more efficient in terms of scalability, autonomy in local attack detection, acceleration on data training near sources and neighbor's parameters sharing. Authors proposed a deep learning approach to detect known and unseen intrusion attacks. Known attacks represent 99% which leads to affirm that zero-day attacks are crafted with small mutations in the old ones. Therefore, multi-layer deep networks enhance small changes awareness (in a self taught algorithm with compression capabilities) compared to shallow learning classifiers. Distributed deep learning approach is based on distributing the dataset to train each sub-dataset locally and rapidly than share and coordinate the learning parameters with neighbors. So the architecture ends with a master IDS which updates the parameters values of the down distributed IDSs and keeps synchronization. The studies show that the distributed parallel deep learning approach realize better results in accuracy than centralized deep learning NIDS and also than shallow machine learning algorithms. To train the models and evaluate the IDS, Diro et al. used NSL-KDD dataset after adding some modification on it to finish with 123 input features and 1 label. As results, they obtained multi-class detection consisting 4 labels (normal, DoS, Probe, R2L.U2R) to achieve 96.5% detection rate and 2.57% of false alarms for deep model in comparison to shallow classifier achieving 93.66% detection and 4.97% false detection rate. They also noted an increase in the overall detection accuracy while adding the number of fog nodes from 96% to 99%. The proposed approach taked longer training time; however, real detection was fast and accurate.

Prabavathy et al. 2018: proposed a novel fog computing based intrusion detection technique using Online Sequential Extreme Learning Machine (OS-ELM). The distributed security mechanism (guaranteed by the fog computing idea) respects interoperability, flexibility, scalability and heterogeneity aspects of IoT systems. The proposed system is composed of the following two major parts:

1) Attack detection at fog nodes: Prabavathy et al. use OSELM algorithm to detect intrusions in fog nodes. The IoT network is divided into virtual clusters where each cluster corresponds to a group of IoT devices under a single fog node. The OS-ELM classifies the incoming packets as normal or an attack. ELM is a single hidden layer feed forward neural network characterized by its fast learning phase. The input layer weights and hidden layer bias values are randomly selected to analytically deduce the output weights using simple matrix computations. However the online nature of OS-ELM favors a streaming detection of IoT attacks.

2) Summarization at cloud server: to have a general idea about the global security state of the IoT system, detected intrusions are sent from the fog node to the cloud server. After the analysis and the visualization of the current state, Prabavathy et al. propose two actions;

i) predict next attacker action using the attacker plan recognition approach; or

ii) identify fog node geographical position based multistage, and DDoS attacks. Hence, an intrusion response can be activated.

He further proposed a proof of concept to evaluate their proposition on DUALCORE processor, 1 GB RAM and 200 GB HDD as fog nodes. Authors deployed Azure cloud service (4 X Dual-Core AMD Opteron 2218 @2.6 GHz, 8 core, 32 GB RAM, 6146 GB HDD) for experimental setup. They implemented OS-ELM using MATLAB and NSL-KDD as benchmark dataset. Authors claimed high accuracy and response time. They achieve 97.36% accuracy with reduced false alarm rate 0.37%. The detection rate with the fog node strategy was 25% faster when compared with cloud based implementation. An important advantage is that new online data can be incorporated in the learning process, which is not the case for ANN and NB.

3.0 Methods

To develop a fingerprint of ransomware's energy consumption, initially, we need to record the power usage of targeted applications. Similar to the approaches in previous studies (Yang 2012; Merlo et al. 2015) we used Power-Tutor to monitor and sample power usage of all running processes in 500 ms intervals. PowerTutor creates log-files containing sequence of energy usage of each process at given sampling interval. We conducted our experiments on three different Android devices, namely: a Samsung Galaxy SIII (CPU: 1.4 GHz, RAM: 2GB, OS: Android 4.4), a Samsung Galaxy S Duos (CPU: 1.0 GHz, RAM: 768 MB, OS: Android 4.0.1), and an Asus Padfone Infinity (CPU: 1.7 GHz, RAM: 2 GB, OS: Android 4.4). To collect energy consumption logs of both ransomware and goodware, we installed the most popular Android applications, namely:

Gmail (version 9.6.83), Facebook (version 99.0.0.26.69),

Google Chrome (version 53.0.2785.124), Youtube (version 11.39.56), Whatsapp (version 2.16.306), Skype (version 7.20.0.411), AngryBrids (version 6.1.5), Google Maps (version 9.39.2), Music Player (version 4.2.52), Twitter (version 6.19.0), Instagram (version 9.6.0) and Guardian (version 3.13.107) and six active and recent ransomware samples on all devices. All ransomware were downloaded via VirusTotal 1 Intelligence API, and these ransomware have active Command and Control (C2) servers.

We then use PowerTutor to monitor and record the device processes' power usage (while running the applications and ransomware, separately) for 5 min. While running the applications (also referred to as goodware), the user interactions mirrored a real world usage. This procedure was repeated five times per device; thus, we obtained

5repeation × 3device = 15 power usage samples for each and every application and ransomware.

As each device's CPU has its own power usage specification, the energy consumption of all devices were mapped to a specific range in order to have a meaningful evaluation. So, we normalised the CPU power consumption for all monitored processes on the devices to [0, 1], where 0 indicates no power usage and 1 presents the maximum CPU power utilisation. Scripts were written to process log-files, extract and normalize power usage values, and generate a row-normalized dataset. Each row includes a label (i.e., goodware or ransomware) and a normalized sequence of energy consumption for five minutes of activity.

3.1 Classification of the Algorithm Used

Assigning correct label to a sample based on previous observations is a key element of Supervised Learning and Classification (Michalski et al. 2013). We applied four state-of-the-art classifiers, namely: k-Nearest Neighbor (KNN),

Neural Network (NN), Support Vector Machine (SVM) and Random Forest (RF), on the power usage samples to recognise the class of each sequence of power consumption. KNN is a simple and powerful classifier which seeks K nearest sample(s) and assigns the majority of neighbor's label to the given samples. NN (Haykin 1998) is an implementation of human brain networks and mostly used to approximate the function between inputs and output.

Another popular technique for supervised learning is SVM, which is based on the concept of decision planes that define decision boundaries. A decision plane differentiates a set of objects based on their class memberships.

Ensemble learning has been the motivation of developing RF (Verikas et al. 2011) that operates by constructing a multitude of decision trees at training time and generating the class label.

Power usage sequence of each process can be considered as time-series data. A wide range of methods have been proposed to classify time-series data (Xing et al. 2010). In this study, a distance based time-series classification approach based on Dynamic Time Warping (DTW) (Müller et al., 2017) is used for distance measure, and KNN is used as a classifier. Similarity distance is a key element in KNN classification and we apply two different distances to find the closest neighbor as follows:

• Euclidean distance: Euclidean distance or Euclidean metric is the intuitive distance between two vectors in Euclidean space and calculated as follow:

• Dynamic time warping (DTW): DTW is a recognized technique for finding an optimal alignment between two time-dependent sequences (see Fig. 1). According to DTW's ability to deal with time deformations and issues associated with speed differences in time-dependent data, it is also employed to calculate distance or similarity between time series (Müller et al., 2017). Let us denote two sequences that display two discrete subsamples as

$X = (x1, \ldots, xn)$ and $Y = (y1, \ldots, ym)$ of length m, $n \epsilon \mathbb{N}$.

DTW uses a Cost Matrix Cϵℝn×m. Each cell Ci,j indicates the distance between xi and yj (see Fig. 2). DTW's purpose is to discover an optimal alignment between X and Y having a minimal entirely distance. As an intuitive explanation, an optimal alignment traverse across a valley of low cost cells within the cost matrix C. A warping path is specified as a sequence p = {p1, ... , pL} with pl = (nl, ml)ϵ[1:N] × [1:M], lϵ[1:L] satisfying the following conditions:

– Boundary condition: p1 = (1, 1) and pL = (N, M).

– Monotonicity condition: n1 ≤ n2 ≤ ⋯ ≤ nL and m1 ≤ m2 ≤ ⋯ ≤ mL.

– Step size condition: pl+1 − pl = {(1, 0), (0, 1), (1, 1)} for l[1:L1].

The summation of all local distances of a warping path's elements outcomes the total cost of path and in order to find optimal warping path p∗, the path having minimum total cost among all possible paths is selected. Finally, to measure similarity or distance between two sequences X and Y, their total cost of optimal warping path are evaluated. The total cost cp(X, Y) of a warping path p between X and Y with respect to the local cost measure c is defined as:

(2)cp(X, Y) =Ll=1

c(xnl, yml).

The DTW distance DTW (X, Y) between X and Y is then defined as the total cost of p∗:

Figure 3 illustrates how DTW aligns two power usage subsamples in order to find optimal path between them for distance calculation.

3.2 Metrics and Cross-Validation

Similar to the approach in (Buczak and Guven 2016), we use the following four common performance indicators for malware detection:

• True positive (TP): indicates that a ransomware is correctly predicted as a malicious application.

• True negative (TN): indicates that a goodware is detected as a non-malicious application correctly.

• False positive (FP): indicates that a goodware is mistakenly detected as a malicious application.

• False negative (FN): indicates that a ransomware is not detected and labelled as a non-malicious application.

To evaluate the effectiveness of our proposed method, we used machine learning performance evaluation metrics that are commonly used in the literature, namely: Accuracy, Recall, Precision and F-Measure.

Accuracy is the number of samples that a classifier correctly detects, divided by the number of all ransomware and goodware applications:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}.$$

Precision is the ratio of predicted ransomware that are correctly labelled a malware. Thus, Precision is defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall or detection rate is the ratio of ransomware samples that are correctly predicted, and is defined as follows:

$$\text{Recall} = \frac{TP}{TP + FN}.$$

F-Measure is the harmonic mean of precision and recall, and is defined as follows:

$$F - \text{Measure} = \frac{2 * TP}{2 * TP + FP + FN}$$

Cross-validation (Kohavi et al. 1995) is a fundamental technique in machine learning to assess the extent that the findings of an experiment can be generalized into an independent dataset. In order to evaluate the performance of the proposed method, we used the leave-one-out cross validation. We are aware that in order to implement this validation method, all subsamples of a sample need to be excluded from the classifier training phase. All evaluations were conducted using MATLAB R2015a running on a Microsoft Windows 10 Pro personal computer powered by Intel Core i7 2.67 GHz and 8 GB RAM.

4.0 Performance Evaluation

Table 2 displays the findings of applying classification algorithms on our dataset. However, as patterns of power consumptions are not predictable and depend on many factors such as files content, encryption algorithm etc. samples are highly distributed in the feature space.

It appears that direct application of conventional classification algorithms namely NN, KNN and SVM, is not promising. For example, the KNN classifier that uses DTW as a similarity measure outperformed other techniques while conventional KNN (with parameter setting of K = 1, 5, 10) is ranked lowest among the classification approaches.

Since Euclidean method calculates similarity by summing distances between corresponding points of samples, the calculated distance could be far when the position of occurring power usage patterns varies (even if samples are visually cognate). On the other hand, DTW attempts to align samples based on the distance between pieces of samples that are more similar regardless of the position of similar energy usage pattern. Consequently, the performance of KNN classifier is significantly influenced by the distance criteria. The second place belongs to RF that selects subset of features and works in splitted feature spaces instead of using a complete feature space. These observations led us to hypothesis that a subset of features (i.e., a specific interval within Ransomware infection period) may improve performance of the classification techniques.

Table 2: performance Evaluation of the Ransomware using different algorithms

Algorithms	Accuracy (%)	Recall (%)	Precision (%)	F-measure (%)
KNN (k = 1)	71.85	71.11	56.14	62.75
KNN (k = 5)	72.59	72.22	57.02	63.73
KNN (k = 10)	83.79	71.11	56.64	63.05
KNN (k = 1 and DTW)	83.79	78.89	73.96	76.34
Neural network	75.93	73.33	61.68	67.01
Random forest	80.74	76.67	69.00	72.63
SVM	78.52	74.44	65.69	69.79

Table 3: Evaluation metrics for different window sizes and SVM: a comparative summary

Data	Accuracy (%)	Recall (%)	Precision (%)	F-measure (%)
5	77.72	59.42	73.21	65.60
10	88.60	85.51	83.10	84.29
15	**91.19**	**94.20**	83.33	**88.44**
20	89.64	82.61	87.69	85.07
25	87.56	75.36	88.14	81.25
30	81.35	55.07	**88.37**	67.86
35	78.24	47.83	84.62	61.11
40	78.24	47.83	84.62	61.11
45	76.17	42.03	82.86	55.77
50	76.68	42.03	85.29	56.31

Best (optimal) values are highlighted in bold

Table 3: Evaluation metrics for different window sizes and neural network: a comparative summary

Data	Accuracy (%)	Recall (%)	Precision (%)	F-measure (%)
5	88.08	82.61	83.82	83.21
10	88.08	84.06	82.86	83.45
15	89.64	**88.41**	83.56	85.92
20	**90.67**	86.96	**86.96**	**86.96**
25	89.64	85.51	85.51	85.51
30	89.12	85.51	84.29	84.89
35	88.08	82.61	83.82	83.21
40	86.01	81.16	80.00	80.58
45	85.49	82.61	78.08	80.28
50	86.01	82.61	79.17	80.85

Best (optimal) values are highlighted in bold

As shown in Table 3, the KNN classifier that uses DTW distance with a subsample

size of 7.5 s outperformed all other methods in terms of detection rate95.65% and performance of 94.27%. Although KNN is the least sophisticated classification approach, it outperformed other rival classification techniques since it only relies on the formation and distribution of goodware's and ransomware's subsamples.

The performance of KNN using DTW for all evaluation metrics peaks at window size = 15. However, the remaining classifiers were not able to achieve an optimal performance at the specified window size. For example, NN's best accuracy, precision and F-measure occurred at w = 20, while highest recall was achieved at w = 15. The numerical results indicate that subsamples are not from specified and exact data distribution and classes have overlap sample(s) in feature

space. Therefore, KNN that seeks for most similar subsample to input data outperform other classification approaches.

Moreover, according to ability to align subsamples, DTW can find closer energy consumption pattern and consequently provide more accurate classification results than euclidean.

Furthermore and in practice, KNN's requirement for concurrent distance calculations between training and testing objects can be implemented using parallel processing (so distances can be independently computed). Subsamples dictionary can be partitioned into sperate IoT nodes and each subsample is sent to nodes. They return a label and a similarity value and the label having less similarity value is final subsample's label. This approach reduces the classification time and mitigates the need for storage capacity in every node.

5.0 Conclusion

With increasing prevalence of Internet-connected devices and things in our data-

centric society, ensuring the security of IoT networks is vital. Successfully compromised IoT nodes could hold the network to ransom adversely affect the operation of an organisation and result in significant financial loss and reputation damage.

In this paper, we presented an approach to detect ransomware, using their power consumption. Specifically, we utilise the unique local fingerprint of ransomware's energy consumption to distinguish ransomware from non-malicious applications. The sequence of applications' energy consumption is splitted into several sequences of power usage subsamples, which are then classified to build aggregated subsample's class labels. Our set of experiments demonstrated that our approach achieved a detection rate of95.65% and a precision rate of 89.19%.

Future works include prototyping the proposed approach for deploying in a real-world IoT network, with the aims of evaluation and refinement.

6.0: References

Ali, B., & Awad, A. I. (2018). Cyber and physical security vulnerability assessment for IoT-based smart homes. *sensors*, *18*(3), 817.

Caceres, R., & Friday, A. (2011). Ubicomp systems at 20: Progress, opportunities, and challenges. *IEEE Pervasive Computing*, *11*(1), 14-21.

Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Generation Computer Systems*, *82*, 761-768.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, *29*(7), 1645-1660.

Karimi, K., & Atkinson, G. (2013). What the Internet of Things (IoT) needs to become a reality. *White Paper, FreeScale and ARM*, 1-16.

Lai, C. F., Lai, Y. X., Yang, L. T., & Chao, H. C. (2012, November). Integration of IoT energy management system with appliance and activity recognition. In *2012 IEEE International Conference on Green*

Computing and Communications (pp. 66-71). IEEE.

Li, Z., Liu, G., Liu, L., Lai, X., & Xu, G. (2017). IoT-based tracking and tracing platform for prepackaged food supply chain. *Industrial Management & Data Systems*.

Merlo, A., Migliardi, M., & Caviglione, L. (2015). A survey on energy-aware security mechanisms. *Pervasive and Mobile Computing*, *24*, 77-90.

Michalski, P. T. (2020). Design and Evaluation of Deadline-Based Scheduling Algorithms for the Industrial Internet of Things.

Müller, J. M., Erdel, M., & Voigt, K. I. (2017). Industry 4.0-Perspectives and challenges for project logistics. In *EurOMA Conference, Edinburgh, Scotland*.

Prabavathy, S., Sundarakantham, K., & Shalinie, S. M. (2018). Design of cognitive fog computing for intrusion detection in Internet of Things. *Journal of Communications and Networks*, *20*(3), 291-298.

Roman, R., Zhou, J., & Lopez, J. (2013). On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, *57*(10), 2266-2279.

Xing, L., Tannous, M., Vokkarane, V. M., Wang, H., & Guo, J. (2017). Reliability modeling of mesh storage area networks for Internet of Things. *IEEE Internet of Things Journal*, *4*(6), 2047-2057.