Using Federated Learning on Malware Classification

Kuang-Yao Lin*, Wei-Ren Huang*

*Cybersecurity Technology Institute, Institute for Information Industry, Taiwan, R.O.C

jasonlin@iii.org.tw, wrhuang@iii.org.tw

***Abstract*— In recent years, everything has been more and more systematic, and it would generate many cyber security issues. One of the most important of these is the malware. Modern malware has switched to a high-growth phase. According to the AV-TEST Institute showed that there are over 350,000 new malicious programs (malware) and potentially unwanted applications (PUA) be registered every day. This threat was presented and discussed in the present paper. In addition, we also considered data privacy by using federated learning. Feature extraction can be performed based on malware. The proposed method achieves very high accuracy (≈0.9167) on the dataset provided by VirusTotal.**

***Keywords*— Malware, Malware family, Classification, Machine learning, Federated learning, Artificial intelligence, Computer security**

I. INTRODUCTION

The AV-TEST Institute registers over 350,000 new malicious programs (malware) and potentially unwanted applications (PUA) every day [1]. Additionally, modern malware is designed with mutation characteristics. According to the “ENISA Threat Landscape Report 2018” 94% of all malicious executables have been polymorphic [2]. As far as the huge number of malwares is concerned, the need for the automation of malware analysis is urgent. However, the metamorphic and polymorphic malwares are a huge challenge for automated malware analysis. In order to solve the challenge, we need a method can depend on the different circumstances, that is artificial intelligence (AI). AI is a term for computer systems that can sense their environment, think, learn, and act in response to what they sense and their programmed objectives [3].

Using AI methods to classify malwares [4, 5, 6, 7] is become more and more popular. All of the methods need collect data into the server, so it would also contain confidential data. Federated learning obtains a central model on the server by aggregating models trained locally on clients. As a result, federated learning does not require clients to upload their data to the server. So, using federated learning will reduce the danger of data transfer. In this paper, we propose a malware classification method by using federated learning. We evaluated our method on the data provided by VirusTotal and achieved 91.67% accuracy.

II. RELATED WORK

Extensive literature exists on classification of malwares. In [4] Ahmadi et al. extracted features from the hex view and the assembly view to classify the malwares that Microsoft released in 2015. In [5] Suarez-Tangil et al. extracted features by resource centric and syntactic to classify the malwares that obtained from McAfee (McGW), Malgenome Project (MgMW) [8], Drebin dataset [9], PRAGuard (PgMW) dataset [10], and malware (MvMW) collected by Lindorfer et al. [11]. In [6] Kinable et al. studied malware classification based on call graph clustering. In [7] Nataraj et al. proposed a malware classification method using image processing techniques. Many studies conclude malware show the importance of malware analysis.

All of the malware analysis methods need collect data into the server. However, Federated learning would change this status. In [12] Li et al. showed that Federated learning can be classified into two categories: global privacy and local privacy. Global privacy requires that the model updates generated at each round are private to all untrusted third parties other than the central server, while local privacy further requires that the updates are also private to the server. So, using federated learning on malware classification would be a good solution.

III.SYSTEM ARCHITECTURE

The most important issue for malware classification is the feature extraction method. Our approach was based on malware binary format with federated learning to find the way to balance speed and accuracy, so we focused on binary content-based features.

A. Dataset

The “ENISA Threat Landscape Report 2018” reported that 79% of the detected malware in organisations were targeting Windows [2]. So, we collect the malware samples for training and testing with Virustotal api, contains the following category on Windows operating system. Our dataset consists of 10,907 malwares includes 5,907 training and 5,000 testing. As shown in Figure 1 and Figure 2, we collected the malwares consists of PUA, Ransom, Trojan, Virus, Backdoor, PWS, SoftwareBundler, TrojanDownloader, VirTool, and Worm.

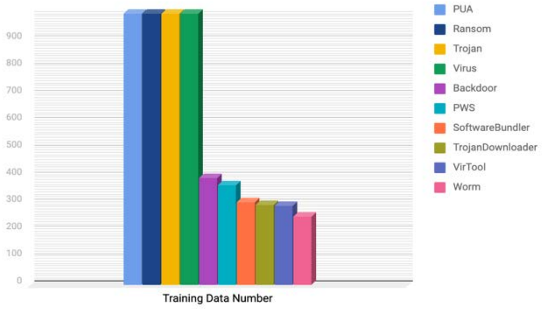


Figure 1. The number of all categories of training data

Authorized licensed use limited to: Temple University. Downloaded on August 10,2021 at 21:32:34 UTC from IEEE Xplore. Restrictions apply.

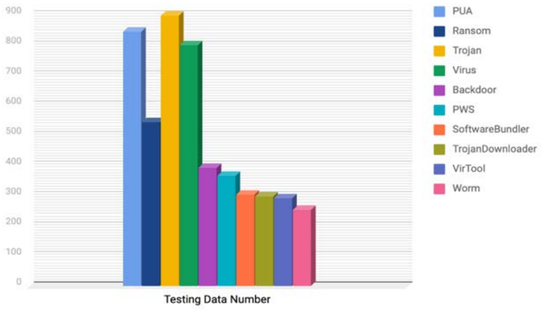


Figure 2. The number of all categories of testing data

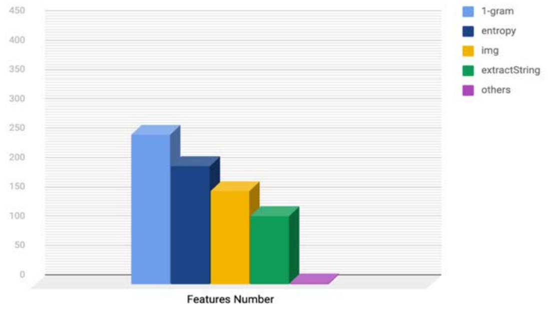


Figure 3. The number of all categories of features

B. *Features*

In the following subsections we provide details on each feature. As shown in Figure 3, there are six kinds of features with a 737-dimensional vector.

1) *1-gram*

An n-gram is a contiguous sequence of n items from a given sample of text or speech. The representation of a malware sample as a sequence of binary values can be effectively described through n-gram analysis to capture information about the type of malware. 1-gram feature which represent the byte frequency and described with a 256-dimensional vector.

2) *Entropy*

Entropy can be defined as a measure of the amount of the disorder and it is used to detect the presence of obfuscation in malware files. For these reasons, we computed the entropy of all the bytes in a malware file and described with a 202-dimensional vector.

3) *Image*

Haralick texture features are common texture descriptors in image analysis. To compute the Haralick features, the image gray-levels are reduced, a process called quantization. Local binary patterns (LBP) is a type of visual descriptor used for classification in computer vision and described with a 160-dimensional vector.

4) *Extract String Length*

Obtained by counting the frequencies at which different string lengths in malware and described with a 116-dimensional vector.

5) *File Size*

File size is a measure of how much data a computer file contains or, alternately, how much storage it consumes, file size is based on byte.

6) *Start Address*

Microsoft migrated to the Portable Executable (PE) format with the introduction of the Windows NT 3.1 operating system. All later versions of Windows, including Windows 95/98/ME and the Win32s addition to Windows 3.1x, support the file structure. Start address is PE file entry point, some malware programs have special starting address.

C. *Models*

We used the several models to test the performance and training speed of malware classification.

1) *SVM*

Support-vector machines (SVM) is supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

2) *LSTM*

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. The malware classification model can be expressed as Figure 4 which with 3 LSTM layers.

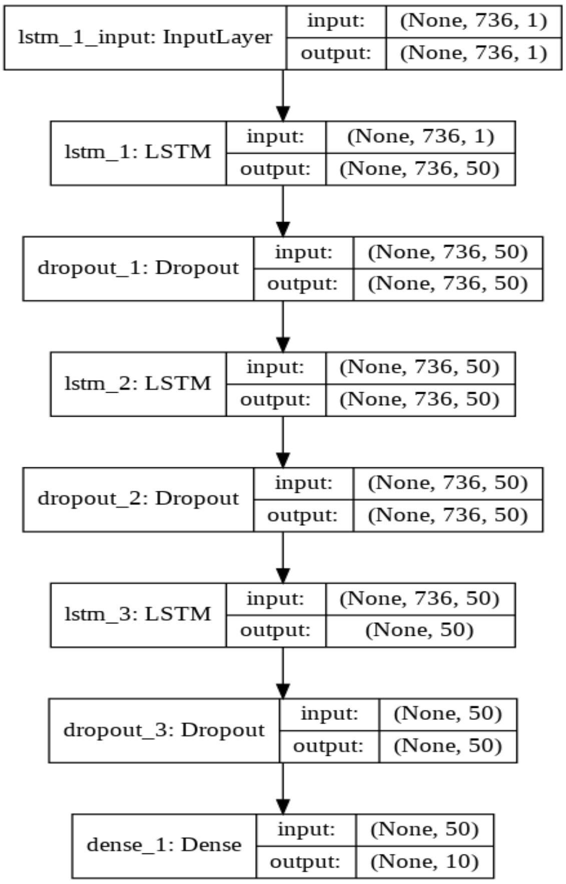


Figure 4. LSTM model

Authorized licensed use limited to: Temple University. Downloaded on August 10,2021 at 21:32:34 UTC from IEEE Xplore. Restrictions apply.

3) *LSTM with Federated Learning*

Considering the problems of malware sharing, we used federated learning to solve multiple decentralized edge devices, without exchanging their data samples. This approach stands in contrast to traditional centralized machine learning techniques where all data samples are uploaded to one server, as well as to more classical decentralized approaches which assume that local data samples are identically distributed.

4) *LSTM + 2 hidden layers with Federated Learning*

We consider how to speed up training without reducing accuracy, use the model show in Figure 5 with one LSTM layer and two full connecting hidden layers.

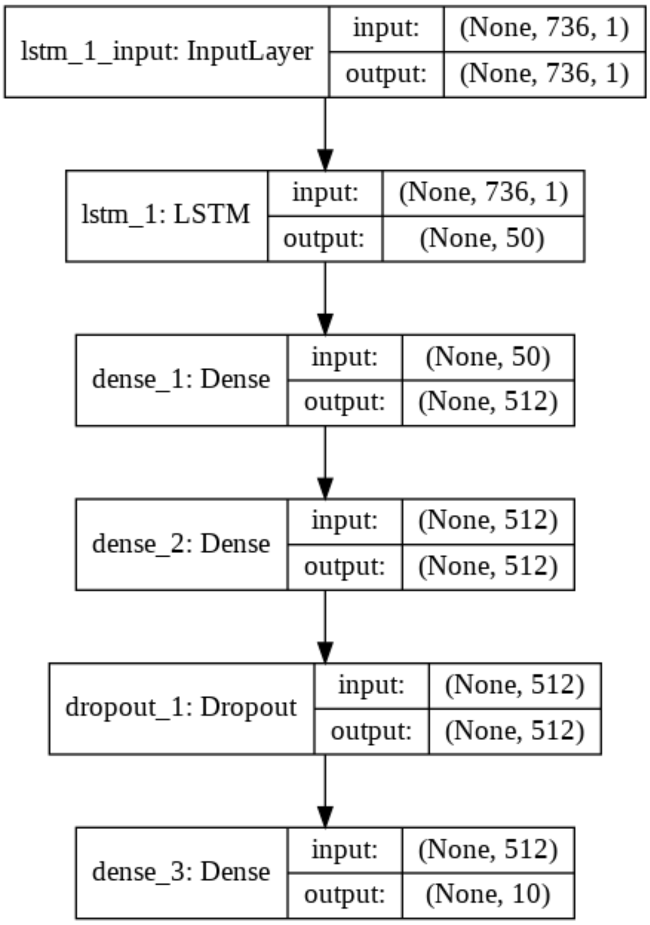


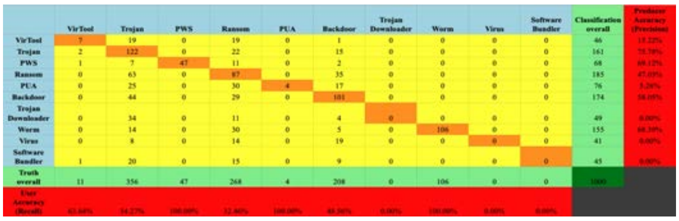
Figure 5. LSTM + 2 hidden layers model

IV. Experiments and Results

We designed several experiments to simulate training scenarios, comparing the effects of machine learning, deep learning, LSTM and full connection. The experimental results are explained below.

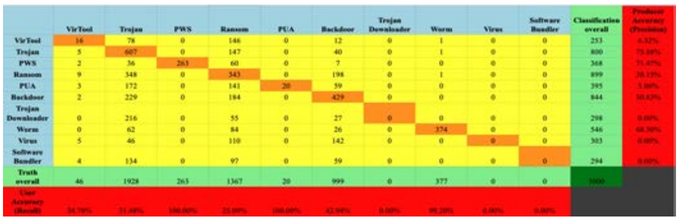
1) *SVM vs Federated learning*

We experimented with 1000 (Figure 6, 7) and 5907 training set (Figure 8, 9), using SVM to find that SVM misclassified on Trojan Downloader, Virus, Software Bundler category with 1000 training set, that is the classification using 1000 training set is not enough for SVM, so we used Federated learning to share training model with different devices.



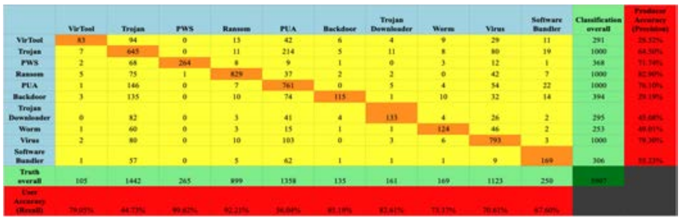
	VirTool	Trojan	PWS	Ransom	PUA	Backdoor	Trojan Downloader	Worm	Virus	Software Bundler	Classification overall	Producer Accuracy (Precision)
VirTool	7	19	0	19	0	1	0	0	0	0	46	15.22%
Trojan	2	122	0	22	0	15	0	0	0	0	161	75.78%
PWS	1	7	47	11	0	2	0	0	0	0	68	69.12%
Ransom	0	63	0	87	0	35	0	0	0	0	185	47.03%
PUA	0	25	0	30	4	17	0	0	0	0	76	5.26%
Backdoor	0	44	0	29	0	101	0	0	0	0	174	58.05%
Trojan Downloader	0	34	0	11	0	4	0	0	0	0	49	0.00%
Worm	0	14	0	30	0	5	0	106	0	0	155	68.39%
Virus	0	8	0	14	0	19	0	0	0	0	41	0.00%
Software Bundler	1	20	0	15	0	9	0	0	0	0	45	0.00%
Truth overall	11	356	47	268	4	208	0	106	0	0	1000	
User Accuracy (Recall)	63.64%	34.27%	100.00%	32.46%	100.00%	48.56%	0.00%	100.00%	0.00%	0.00%		

Figure 6. confusion matrix of SVM with 1000 training data



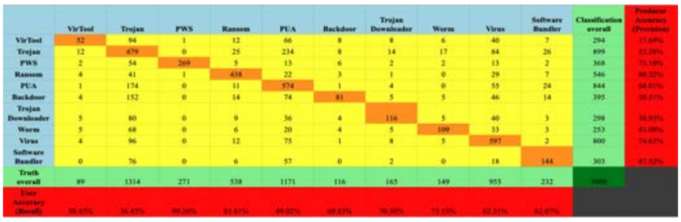
	VirTool	Trojan	PWS	Ransom	PUA	Backdoor	Trojan Downloader	Worm	Virus	Software Bundler	Classification overall	Producer Accuracy (Precision)
VirTool	16	78	0	146	0	12	0	1	0	0	253	6.32%
Trojan	5	607	0	147	0	40	0	1	0	0	800	75.88%
PWS	2	36	263	60	0	7	0	0	0	0	368	71.47%
Ransom	9	348	0	343	0	198	0	1	0	0	899	38.15%
PUA	3	172	0	141	20	59	0	0	0	0	395	5.06%
Backdoor	2	229	0	184	0	429	0	0	0	0	844	50.83%
Trojan Downloader	0	216	0	55	0	27	0	0	0	0	298	0.00%
Worm	0	62	0	84	0	26	0	374	0	0	546	68.50%
Virus	5	46	0	110	0	142	0	0	0	0	303	0.00%
Software Bundler	4	134	0	97	0	59	0	0	0	0	294	0.00%
Truth overall	46	1928	263	1367	20	999	0	377	0	0	5000	
User Accuracy (Recall)	34.78%	31.48%	100.00%	25.09%	100.00%	42.94%	0.00%	99.20%	0.00%	0.00%		

Figure 7. confusion matrix of SVM with 1000 training data to predict testing data



	VirTool	Trojan	PWS	Ransom	PUA	Backdoor	Trojan Downloader	Worm	Virus	Software Bundler	Classification overall	Producer Accuracy (Precision)
VirTool	83	94	0	13	42	6	4	9	29	11	291	28.52%
Trojan	7	645	0	11	214	5	11	8	80	19	1000	64.50%
PWS	2	68	264	8	9	1	0	3	12	1	368	71.74%
Ransom	5	75	1	829	37	2	2	0	42	7	1000	82.90%
PUA	1	146	0	7	761	0	5	4	54	22	1000	76.10%
Backdoor	3	135	0	10	74	115	1	10	32	14	394	29.19%
Trojan Downloader	0	82	0	3	41	4	133	4	26	2	295	45.08%
Worm	1	60	0	3	15	1	1	124	46	2	253	49.01%
Virus	2	80	0	10	103	0	3	6	793	3	1000	79.30%
Software Bundler	1	57	0	5	62	1	1	1	9	169	306	55.23%
Truth overall	105	1442	265	899	1358	135	161	169	1123	250	5907	
User Accuracy (Recall)	79.05%	44.73%	99.62%	92.21%	56.04%	85.19%	82.61%	73.37%	70.61%	67.60%		

Figure 8. confusion matrix of SVM with 5907 training data

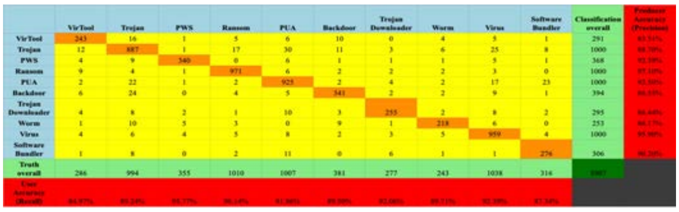


	VirTool	Trojan	PWS	Ransom	PUA	Backdoor	Trojan Downloader	Worm	Virus	Software Bundler	Classification overall	Producer Accuracy (Precision)
VirTool	52	94	1	12	66	8	8	6	40	7	294	17.69%
Trojan	12	479	0	25	234	8	14	17	84	26	899	53.28%
PWS	2	54	269	5	13	6	2	2	13	2	368	73.10%
Ransom	4	41	1	438	22	3	1	0	29	7	546	80.22%
PUA	1	174	0	11	574	1	4	0	55	24	844	68.01%
Backdoor	4	152	0	14	74	81	5	5	46	14	395	20.51%
Trojan Downloader	5	80	0	9	36	4	116	5	40	3	298	38.93%
Worm	5	68	0	6	20	4	5	109	33	3	253	43.08%
Virus	4	96	0	12	75	1	8	5	597	2	800	74.63%
Software Bundler	0	76	0	6	57	0	2	0	18	144	303	47.52%
Truth overall	89	1314	271	538	1171	116	165	149	955	232	5000	
User Accuracy (Recall)	58.43%	36.45%	99.26%	81.41%	49.02%	69.83%	70.30%	73.15%	62.51%	62.07%		

Figure 9. confusion matrix of SVM with 5907 training data to predict testing data

2) *Traditional learning vs Federated learning*

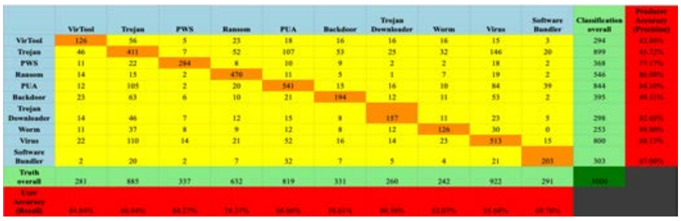
In the traditional learning methods sensitive training data are collected together where models are trained, we discovered the data of each team, group, company is difficult to share. Federated learning is a decentralized model, born at the intersection of on-device and edge computing. In contrast to the traditional learning methods, Federated Learning brings the models to the data source or client device for training and inferencing.



	VirTool	Trojan	PWS	Ransom	PUA	Backdoor	Trojan Downloader	Worm	Virus	Software Bundler	Classification overall	Producer Accuracy (Precision)
VirTool	243	16	1	5	6	10	0	4	5	1	291	83.51%
Trojan	12	887	1	17	30	11	3	6	25	8	1000	88.70%
PWS	4	9	340	0	6	1	1	1	5	1	368	92.39%
Ransom	9	4	1	971	6	2	2	2	3	0	1000	97.10%
PUA	2	22	1	2	925	2	4	2	17	23	1000	92.50%
Backdoor	6	24	0	4	5	341	2	2	9	1	394	86.55%
Trojan Downloader	4	8	2	1	10	3	255	2	8	2	295	86.44%
Worm	1	10	5	3	0	9	1	218	6	0	253	86.17%
Virus	4	6	4	5	8	2	3	5	959	4	1000	95.90%
Software Bundler	1	8	0	2	11	0	6	1	1	276	306	90.20%
Truth overall	286	994	355	1010	1007	381	277	243	1038	316	5907	
User Accuracy (Recall)	84.97%	89.24%	95.77%	96.14%	91.86%	89.50%	92.06%	89.71%	92.39%	87.34%		

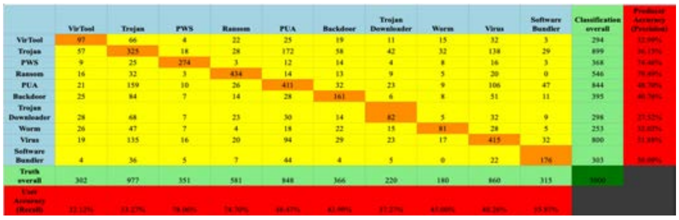
Figure 10. confusion matrix of federated learning on training data (10000 epoch)

Authorized licensed use limited to: Temple University. Downloaded on August 10,2021 at 21:32:34 UTC from IEEE Xplore. Restrictions apply.



	VirTool	Trojan	PWS	Ransom	PUA	Backdoor	Trojan Downloader	Worm	Virus	Software Bundler	Classification overall	Producer Accuracy (Precision)
VirTool	126	56	5	23	18	16	16	16	15	3	294	42.86%
Trojan	46	411	7	52	107	53	25	32	146	20	899	45.72%
PWS	11	22	284	8	10	9	2	2	18	2	368	77.17%
Ransom	14	15	2	470	11	5	1	7	19	2	546	86.08%
PUA	12	105	2	20	541	15	16	10	84	39	844	64.10%
Backdoor	23	63	6	10	21	194	12	11	53	2	395	49.11%
Trojan Downloader	14	46	7	12	15	8	157	11	23	5	298	52.68%
Worm	11	37	8	9	12	8	12	126	30	0	253	49.80%
Virus	22	110	14	21	52	16	14	23	513	15	800	64.13%
Software Bundler	2	20	2	7	32	7	5	4	21	203	303	67.00%
Truth overall	281	885	337	632	819	331	260	242	922	291	5000	
User Accuracy (Recall)	44.84%	46.44%	84.27%	74.37%	66.06%	58.61%	60.38%	52.07%	55.64%	69.76%		

Figure 11. confusion matrix of federated learning on testing data (10000 epoch)

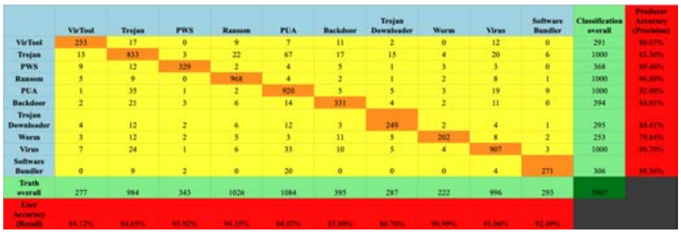


	VirTool	Trojan	PWS	Ransom	PUA	Backdoor	Trojan Downloader	Worm	Virus	Software Bundler	Classification overall	Producer Accuracy (Precision)
VirTool	97	66	4	22	25	19	11	15	32	3	294	32.99%
Trojan	57	325	18	28	172	58	42	32	138	29	899	36.15%
PWS	9	25	274	3	12	14	4	8	16	3	368	74.46%
Ransom	16	32	3	434	14	13	9	5	20	0	546	79.49%
PUA	21	159	10	26	411	32	23	9	106	47	844	48.70%
Backdoor	25	84	7	14	28	161	6	8	51	11	395	40.76%
Trojan Downloader	28	68	7	23	30	14	82	5	32	9	298	27.52%
Worm	26	47	7	4	18	22	15	81	28	5	253	32.02%
Virus	19	135	16	20	94	29	23	17	415	32	800	51.88%
Software Bundler	4	36	5	7	44	4	5	0	22	176	303	58.09%
Truth overall	302	977	351	581	848	366	220	180	860	315	5000	
User Accuracy (Recall)	32.12%	33.27%	78.06%	74.70%	48.47%	43.99%	37.27%	45.00%	48.26%	55.87%		

Figure 12. confusion matrix of traditional learning on testing data (10000 epoch)

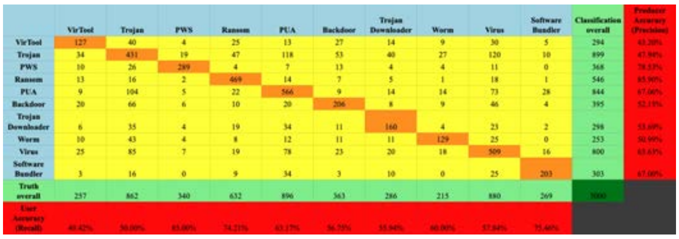
3) *LSTM vs LSTM + 2 hidden layers*

How to train a model in a limited time, the first approach is solved by hardware, and the second approach is solved by model complexity. Under the same hardware conditions, it can only be processed by reducing the model complexity. After many experiments, we can get a good accuracy by using only one layer of LSTM, compared with the 3 LSTM model, it can be reduced to 1/3 training time, so we can obtain more powerful models in limited time.



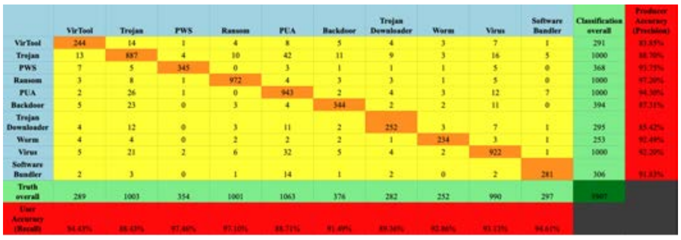
	VirTool	Trojan	PWS	Ransom	PUA	Backdoor	Trojan Downloader	Worm	Virus	Software Bundler	Classification overall	Producer Accuracy (Precision)
VirTool	233	17	0	9	7	11	2	0	12	0	291	80.07%
Trojan	13	833	3	22	67	17	15	4	20	6	1000	83.30%
PWS	9	12	329	2	4	5	1	3	3	0	368	89.40%
Ransom	5	9	0	968	4	2	1	2	8	1	1000	96.80%
PUA	1	35	1	2	920	5	5	3	19	9	1000	92.00%
Backdoor	2	21	3	6	14	331	4	2	11	0	394	84.01%
Trojan Downloader	4	12	2	6	12	3	249	2	4	1	295	84.41%
Worm	3	12	2	5	3	11	5	202	8	2	253	79.84%
Virus	7	24	1	6	33	10	5	4	907	3	1000	90.70%
Software Bundler	0	9	2	0	20	0	0	0	4	271	306	88.56%
Truth overall	277	984	343	1026	1084	395	287	222	996	293	5907	
User Accuracy (Recall)	84.12%	84.65%	95.92%	94.35%	84.87%	83.80%	86.76%	90.99%	91.06%	92.49%		

Figure 13. confusion matrix of 3 LSTM model on training data (1000 epoch)



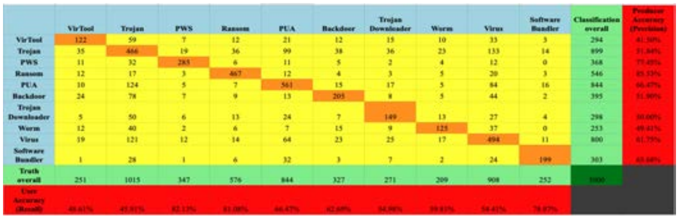
	VirTool	Trojan	PWS	Ransom	PUA	Backdoor	Trojan Downloader	Worm	Virus	Software Bundler	Classification overall	Producer Accuracy (Precision)
VirTool	127	40	4	25	13	27	14	9	30	5	294	43.20%
Trojan	34	431	19	47	118	53	40	27	120	10	899	47.94%
PWS	10	26	289	4	7	13	4	4	11	0	368	78.53%
Ransom	13	16	2	469	14	7	5	1	18	1	546	85.90%
PUA	9	104	5	22	566	9	14	14	73	28	844	67.06%
Backdoor	20	66	6	10	20	206	8	9	46	4	395	52.15%
Trojan Downloader	6	35	4	19	34	11	160	4	23	2	298	53.69%
Worm	10	43	4	8	12	11	11	129	25	0	253	50.99%
Virus	25	85	7	19	78	23	20	18	509	16	800	63.63%
Software Bundler	3	16	0	9	34	3	10	0	25	203	303	67.00%
Truth overall	257	862	340	632	896	363	286	215	880	269	5000	
User Accuracy (Recall)	49.42%	50.00%	85.00%	74.21%	63.17%	56.75%	55.94%	60.00%	57.84%	75.46%		

Figure 14. confusion matrix of 3 LSTM model on testing data (1000 epoch)



	VirTool	Trojan	PWS	Ransom	PUA	Backdoor	Trojan Downloader	Worm	Virus	Software Bundler	Classification overall	Producer Accuracy (Precision)
VirTool	244	14	1	4	8	5	4	3	7	1	291	83.85%
Trojan	13	887	4	10	42	11	9	3	16	5	1000	88.70%
PWS	7	5	345	0	3	1	1	1	5	0	368	93.75%
Ransom	3	8	1	972	4	3	3	1	5	0	1000	97.20%
PUA	2	26	1	0	943	2	4	3	12	7	1000	94.30%
Backdoor	5	23	0	3	4	344	2	2	11	0	394	87.31%
Trojan Downloader	4	12	0	3	11	2	252	3	7	1	295	85.42%
Worm	4	4	0	2	2	2	1	234	3	1	253	92.49%
Virus	5	21	2	6	32	5	4	2	922	1	1000	92.20%
Software Bundler	2	3	0	1	14	1	2	0	2	281	306	91.83%
Truth overall	289	1003	354	1001	1063	376	282	252	990	297	5907	
User Accuracy (Recall)	84.43%	88.43%	97.46%	97.10%	88.71%	91.49%	89.36%	92.86%	93.13%	94.61%		

Figure 15. confusion matrix of LSTM + 2 hidden layer model on training data (1000 epoch)



	VirTool	Trojan	PWS	Ransom	PUA	Backdoor	Trojan Downloader	Worm	Virus	Software Bundler	Classification overall	Producer Accuracy (Precision)
VirTool	122	59	7	12	21	12	15	10	33	3	294	41.50%
Trojan	35	466	19	36	99	38	36	23	133	14	899	51.84%
PWS	11	32	285	6	11	5	2	4	12	0	368	77.45%
Ransom	12	17	3	467	12	4	3	5	20	3	546	85.53%
PUA	10	124	5	7	561	15	17	5	84	16	844	66.47%
Backdoor	24	78	7	9	13	205	8	5	44	2	395	51.90%
Trojan Downloader	5	50	6	13	24	7	149	13	27	4	298	50.00%
Worm	12	40	2	6	7	15	9	125	37	0	253	49.41%
Virus	19	121	12	14	64	23	25	17	494	11	800	61.75%
Software Bundler	1	28	1	6	32	3	7	2	24	199	303	65.68%
Truth overall	251	1015	347	576	844	327	271	209	908	252	5000	
User Accuracy (Recall)	48.61%	45.91%	82.13%	81.08%	66.47%	62.69%	54.98%	59.81%	54.41%	78.97%		

Figure 16. Confusion matrix of LSTM + 2 hidden layer model on testing data (1000 epoch)

V. Conclusions

Cyber security and data privacy are becoming more and more necessary. So, we presented a malware classification prototype characterized by a malware classification method with decentralized data collection. To attain this goal, we proposed a concept using federated learning on malware classification. We showed the experiments and results by compared SVM and LSTM. In addition, we also used various combinations of algorithm. In the experiment, we discovered that malware sharing can achieve a high accuracy. Additionally, the solution by decentralized or reducing the model complexity could also cause an acceptable result. Therefore, using federated learning on malware classification would be a great solution.

Acknowledgment

This research is supported by industry and institute in Taiwan, and the authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the contributions of this paper.

References

[1] "Malware Statistics & Trends Report: AV-TEST," AV-TEST, 25-Apr-2019. [Online]. Available: https://www.av-test.org/en/statistics/malware/. [Accessed: 27-Dec-2019].

[2] ENISA Threat Landscape Report 2018: 15 Top Cyber-Threats and Trends. ENISA, 2019.

[3] "Harnessing Artificial Intelligence for the Earth," Fourth Industrial Revolution for the Earth series. [online]. Available: https://www.pwc.com/gx/en/sustainability/assets/ai-for-the-earth-jan-2018.pdf. [Accessed: 27-Dec-2019].

[4] M. Ahmadi, D. Ulyanov, S. Semenov, M. Trofimov, and G. Giacinto, "Novel Feature Extraction, Selection and Fusion for Effective Malware Family Classification," Proceedings of the Sixth ACM on Conference on Data and Application Security and Privacy - CODASPY 16, 2016.

[5] G. Suarez-Tangil, S. K. Dash, M. Ahmadi, J. Kinder, G. Giacinto, and L. Cavallaro, "DroidSieve: Fast and accurate classification of obfuscated android malware," Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy - CODASPY 17, 2017.

[6] J. Kinable and O. Kostakis, "Malware classification based on call graph clustering," Journal in Computer Virology, vol. 7, no. 4, pp. 233–245, Mar. 2011.

[7] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware Images: Visualization and Automatic Classification," Proceedings of the 8th International Symposium on Visualization for Cyber Security - VizSec 11, 2011.

[8] Y. Zhou and X. Jiang, "Dissecting Android Malware: Characterization and Evolution," 2012 IEEE Symposium on Security and Privacy, 2012.

[9] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, and K. Rieck, "Drebin: Effective and Explainable Detection of Android Malware in Your Pocket," Proceedings 2014 Network and Distributed System Security Symposium, 2014.

Authorized licensed use limited to: Temple University. Downloaded on August 10,2021 at 21:32:34 UTC from IEEE Xplore. Restrictions apply.

[10] D. Maiorca, D. Ariu, I. Corona, M. Aresu, and G. Giacinto, “Stealth attacks: An extended insight into the obfuscation effects on Android malware,” Computers & Security, vol. 51, pp. 16–31, 2015.

[11] M. Lindorfer, M. Neugschwandtner, and C. Platzer, “MARVIN: Efficient and Comprehensive Mobile App Classification through Static and Dynamic Analysis,” 2015 IEEE 39th Annual Computer Software and Applications Conference, 2015.

[12] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” arXiv preprint arXiv:1908.07873, Aug. 2019.

Kuang-Yao Lin received the B.S. degree in information management from Chang Gung University, Taoyuan, Taiwan, R.O.C, in 2015. He received the M.S. degree in information management from Chang Gung University, Taoyuan, Taiwan, R.O.C, in 2017. He is currently an engineer in Cybersecurity Technology Institute, Institute for information industry, Taiwan, R.O.C. His research interests include machine learning, malware analysis, system design and development and cyber security.

Wei-Ren Huang received the B.S. degree in mathematics and information education from National Taipei University of Education, Taipei, Taiwan, R.O.C, in 2007. He received the M.S. degree in multimedia engineering from National Chiao Tung University, Hsinchu, Taiwan, R.O.C, in 2009. He is currently an engineer in Cybersecurity Technology Institute, Institute for information industry, Taiwan, R.O.C. His research interests include machine learning, data mining, malware analysis, system design and development and cyber security.

Authorized licensed use limited to: Temple University. Downloaded on August 10,2021 at 21:32:34 UTC from IEEE Xplore. Restrictions apply.