2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering

Identifying WeChat Red Packets and Fund Transfers via Analyzing Encrypted Network Traffic

Feipeng Yan*, Ming Xu†*, Tong Qiao✉†, Ting Wu†, Xue Yang*, Ning Zheng†*, Kim-Kwang Raymond Choo‡
*School of Computer Science and Technology, Hangzhou Dianzi University, HangZhou, China
†School of Cyberspace, Hangzhou Dianzi University, HangZhou, China
‡Department of Information Systems and Cyber Security, The University of Texas at San Antonio, TX 78249, San Antonio, USA

***Abstract*— WeChat is an extremely popular application in China and among the overseas Chinese users, and two widely used features are giving of red packet (a Chinese customary practice of giving money in red envelope) or fund transfer. Investigation of WeChat red packet and fund transfer transactions is an understudied topic, and hence the focus of this paper. Specifically, we analyze the encrypted network traffic involving WeChat red packet and fund transfer transactions. We segment the traffic into several bursts describing the different actions. Then, we extract relevant red packet transaction and fund transfer features from each burst, which are then used to train a learning-based classifier to distinguish between the different bursts. The findings from our evaluation demonstrate that our proposed approach can accurately identify the actions of red packet transactions and fund transfers, as well as accurately predicting the number of red packet transactions and fund transfers.**

***Index Terms*—Traffic classification, WeChat analysis, Network traffic analysis, User action identification**

I. Introduction

WeChat is the largest social messaging platform in China, offering features such as sending/receiving of voice messages, videos, pictures, texts, red packets[1], and fund transfers[2]. Among those actions, red packet and fund transfers features allow users to send or receive virtual money online. According to Tencent (the company that developed and maintained WeChat), the number of the sent red packets over WeChat on Lunar New Year's Eve of 2017 reached 600,000 per second.

Similar to other popular consumer technologies, there exist a large scale of privacy and security risks to the users. For instance, a malicious sender may send an image depicting red packets or fund transfers, but containing a phishing URL attempting to defraud an unwitting receiver. One particular line of research is identifying user actions. However, user action identification via network traffic analysis, for example by investigating the payload's content, is not new.

Increasingly, mobile applications (apps) are encrypting the content of payloads; thus, rendering network traffic analysis ineffective. Machine learning methods have also been used to classify apps or user actions, based on the analysis of flow statistical features (also known as side-channel information) [1], [2]. Since the features are both port-and payload-independent, such approaches are not affected by encrypted traffic and can be used for user action identification [3]–[8].

Watkins et al. [3], for example, utilized the packet inter-arrival time of responses to Internet control message protocol (ICMP) packets, to predict user actions. The authors in [4] attempted to identify the action a user executed on some apps by analyzing the encrypted network traffic. In [5], WeChat texts and pictures were classified using four known machine learning techniques via analyzing WeChat encrypted traffic. The authors in [6] considered 11 actions in KakaoTalk, and learned its traffic pattern by recognizing the specific action in the hidden network traffic. In [7], the authors analyzed the sizes of packets exchanged between the target user and Apple's server in their attempt to identify user actions. A system for classifying service usages via the analysis of encrypted network traffic was presented in [8]. In this approach, network traffic was segmented hierarchically in order to describe the characteristics of different user actions.

Mobile app security, privacy and forensics are a topic of ongoing interest [9], [10]. For example, Yan et al. [11] proposed a system to identify sensitive user input which requires further protection. Their approach is based on both machine learning and program analysis techniques. Christoforos et al. [12] and Farden, Martini and Choo [13] demonstrated that it is possible to recover authentication credentials / tokens of Android mobile apps from the devices. Shetty et al. [14] devised an adversary model based approach for analyzing security vulnerabilities and designing weakness in mobile dating apps. They demonstrated how to use man-in-middle attack against most of dating apps. In addition, if an attacker has physical access to devices where the apps are installed, it could also be possible to forensically recover authentication credentials / tokens and other sensitive user information, as demonstrated in [15]–[17]. The authors of [18] proposed a method to infer the type of mobile device used (e.g. Android and iOS devices) by analyzing the traffic generated. Costa et al. [19] attempted to validate the security of a group of mobile

✉ Corresponding author: Tong Qiao (*E-mail: tong.qiao@hdu.edu.cn*).

[1]WeChat red packet is also referred to as "Lucky Money" or "Red Envelope".

[2]WeChat fund transfer allows transfer of funds between individual users and small businesses.

2324-9013/18/31.00 ©2018 IEEE
DOI 10.1109/TrustCom/BigDataSE.2018.00198

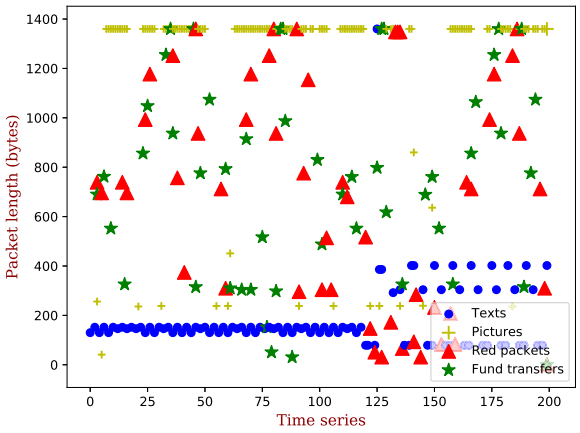


Fig. 1: Estimated packet lengths for texts, pictures, red packet, and fund transfers.

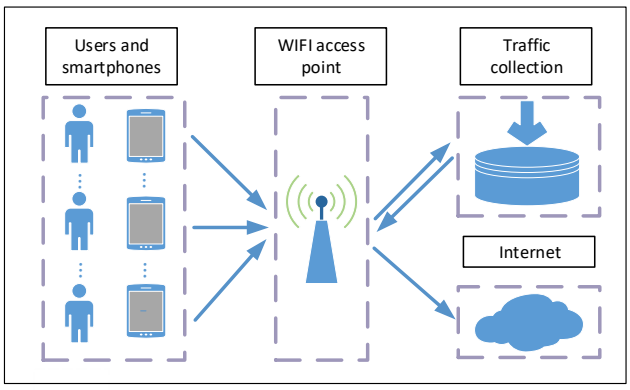


Fig. 2: Proposed traffic collection platform for WeChat.

apps. In [20], the authors designed a mobile app that captures and investigates the security policy of the cellular operator. The app is designed to extract relevant parameters such as encryption keys and locations of users.

There have also been some interest in identifying different actions on WeChat in recent years. Specifically, the authors in [5], [8] studied the sending / receiving of texts or pictures by analyzing the encrypted network traffic. However, we are not aware of any existing literature studying red packet and fund transfer transactions in WeChat. The potential of red packet and fund transfers features being abused for criminal activities, such as gambling, money laundering and terrorism financing, necessitates the study of both features from both security and forensic perspectives.

Hence, in this paper, we study the security of this app, focusing on red packet and fund transfer transactions. Specifically, we find that the traffic pattern generated by red packet and fund transfer differ from other actions. Thus, using the machine learning classifier proposed in this paper to analyze the encrypted WeChat network traffic, we demonstrate how we can identify a malicious activity (e.g. phishing attempts or online gambling activities) disguised as a red packet / fund transfer, as well as accurately predicting the number of red packet transactions and fund transfers.

In the next section, we describe the characteristics of red packet and fund transfer. In Sections III and IV, we present our classification system and its evaluation, respectively. Finally, let us conclude the paper and discuss potential future work in Section V.

II. Characteristics of Red Packet and Fund Transfers

The traffic associated with red packet and fund transfer transactions may have the following differences:

- The packet length (filtered out the zero-payload packet) of red packet and fund transfer transactions is likely to differ from that of text and pictures (see Fig. 1). Comparing with typical text, red packets or fund transfers carry additional information such as the amount, users' balance, password, and fingerprint. For example, we estimated that nearly 80% of the packet length of pictures and text is distributed around 1300 bytes and 180 bytes, respectively. However, 70% of the packet length of red packet and fund transfer transactions range from 300 bytes to 1300 bytes, and more discrete than those of text or pictures.
- Sending / receiving of red packet and fund transfers consists of the steps described in Table I (action-1 represents sending / receiving text; action-2 represents sending / receiving picture), which result in a distinguishable network traffic pattern. For instance, a sender firstly clicks the function of red packet when a sender wants to send red packet to other users, and next fills in the account, and ready to pay, and then pays for it by password or fingerprint. The sender will receive a message when the receiver opens the red packet.
- Since both red packet and fund transfer transactions involve cash flow, users have to interact with the WeChat server repeatedly, resulting in more handshakes than texts and pictures. Because most step of red packets and fund transfers will produce several TCP handshakes.

In the approaches proposed in [5], [8], the traffic associated with the different steps of a red packet or fund transfer is not being separated; thus, red packet and fund transfer transactions cannot be effectively classified. Besides, the number of these actions remains unknown. To deal with this limitation, we segment the traffic into several bursts, each representing a typical action (i.e. sending / receiving of a text or a picture, the individual step in the sending / receiving of a red packet or fund transfer). Then, we design a classification system to identify red packet and fund transfer transactions, as well as estimating the number of red packet and fund transfer transactions.

TABLE I: Steps involved in red packet and fund transfer transactions

Step	Sending a red packet	Receiving a red packet	Sending a fund transfer	Receiving a fund transfer
i	clicks the function of red packet (action-3)	receives a red packet (action-7)	clicks the function of fund transfer (action-10)	receives a fund transfer (action-14)
ii	fills in the account and ready to pay (action-4)	clicks the red packet (action-8)	fills in the account and ready to pay (action-11)	clicks on the fund transfer (action-15)
iii	pays for it by password or fingerprint (action-5)	opens the red packet (action-9)	pays for it by password or fingerprint (action-12)	confirms the fund transfer (action-16)
vi	receiver opens the red packet (action-6)	/	receiver confirms the fund transfer (action-13)	/

In the next section, we will present our proposed classification system.

III. Proposed Classification System

The general framework of our proposed system comprises four modules, namely:

1) traffic collection (see Section III-A);
2) traffic segmentation (see Section III-B);
3) features extraction (see Section III-C); and
4) classifier establishment (see Section III-D).

A. Traffic Collection

The four basic WeChat actions are the sending / receiving of texts, pictures, red packets, and fund transfers. For the latter two actions, there are additional steps involved, as outlined in Table I.

Thus, in this work, we propose collecting the network traffic by monitoring user actions on WeChat. Fig. 2 depicts our traffic collection platform, where users use the WeChat app from a range of mobile devices (e.g. Android and iOS devices) that are connected to same access point. WeChat can be used for different activities, and the timestamp of each of these activities / actions is recorded. There are many open source tools that can be used to collect such traffic, and one popular tool is Wireshark.

To avoid "contaminating" the network traffic, we limit the networking capabilities of all other apps installed on the device used in our evaluation. In other words, only WeChat was generating the network traffic.

B. Traffic Segmentation

Traffic segmentation includes the following three steps:

1) **Traffic pre-processing**: We filter irrelevant packets. For instance, for missing, damaged, duplicated packages, and packages that are delivered out of order, TCP will request the retransmission of such problematic data. Besides, the ACK packets do not carry data between communication, as it indicates that the data have been received correctly. We also consider the potential presence of abnormal traffic. For instance, even though we limit the networking capabilities of other apps, the operating system may still generate traffic that we are not able to block. Therefore, we need to discard abnormal traffic, based on the record of all actions.

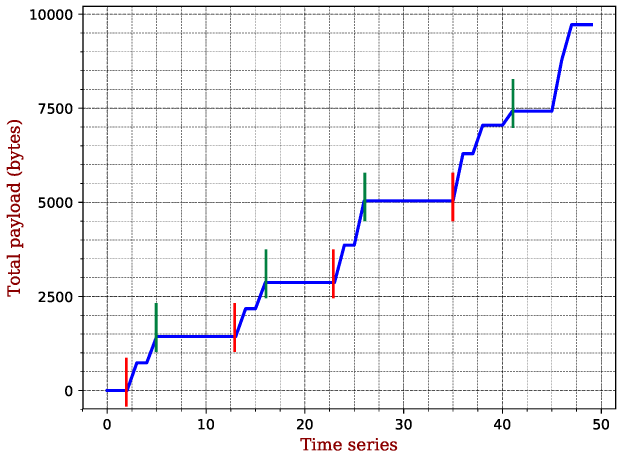


Fig. 3: The descriptions of burst.

2) **Time series transformation**: We transform packets into time series containing side-channel information: packet length, timestamp and TCP flag. Packet length denotes the length of each action. The inter-packet time can be calculated by timestamp, and it is used to segment traffic into bursts. TCP flags can be used to determine whether the packet serves as a TCP handshake.
3) **Segmenting traffic into bursts**: A burst can be defined as a group of consecutive packets, where two consecutive packets occur within the threshold time period (or burst threshold). For instance, if two consecutive packets do not occur within a burst threshold, then these packets are segmented into different bursts. Fig. 3 illustrates the change in total payload along time series. According to the definition of burst, the starting point (red line) of a burst can be recorded as the slope of the curve in Fig. 3 is greater than zero, while the ending point (green line) can be labeled when the slope equals to zero. The time span between the starting point and ending point implies that large amounts of data generated by user actions are constantly transmitted. On the contrary, no data is transmitted when the slope is continually equates zero.
Accordingly, the duration of time span plays an important role in the selection of burst threshold. For instance,

the different types of traffic generated by different user actions might be segmented by the same burst if the inappropriate burst threshold is selected. It practice, 95% of packets on smartphones are received or transmitted within one second (time span) of the previous packet in most current network environment (see [21] for instance). Thus, the selection of burst threshold is limited into the range from zero to two seconds. Besides, the selection of the burst threshold will be elaborated in Section IV-C.

C. Feature Extraction

After segmenting the traffic into bursts, we extract features from each burst based on its time series.

Overall statistics: We use statistical features to describe the basic properties of burst. It is proposed to collect a sequence with each packet length in the same burst. Then, the first order and second order descriptive statistics are obtained, which include sum, mean, standard deviation, skewness, and kurtosis.

Packet length: We use packet length to describe the overall distribution. Different actions result in different (ranges of) packet lengths. Let us divide the range into several equal-sized sub-ranges in 300-byte steps, and calculate the number of packets in each sub-range.

Number of TCP handshakes: WeChat red packet and fund transfer transactions always produce different number of TCP handshakes with other actions, and a packet of TCP handshake can be identified if the TCP flag is labeled as "SYN", "SYN+ACK", "FIN" or "FIN+ACK".

Inbound and outbound statistics: The features (inbound and outbound statistics) describe the properties of the different directions. Different directions have different characteristics, such as the total payloads and the number of packets. For example, the total payload of sending a red packet is different from receiving a red packet.

D. Classifier Establishment

After extracting the features, we can acquire the vectors representing bursts labeled as the training set and the testing set. We leverage the vectors to train the ensemble classifier, which is used to distinguish the actions represented by different bursts. We also propose comparing different machine learning methods to train the classifier. In this paper, we use the classifier based on Random Forest (RF) [22] and compare the performance using different number of decision trees in RF. Based on the findings reported in Fig. 5 and Fig. 6, we select the optimal algorithm.

IV. Evaluation

In this section, we will describe our evaluation approach and findings.

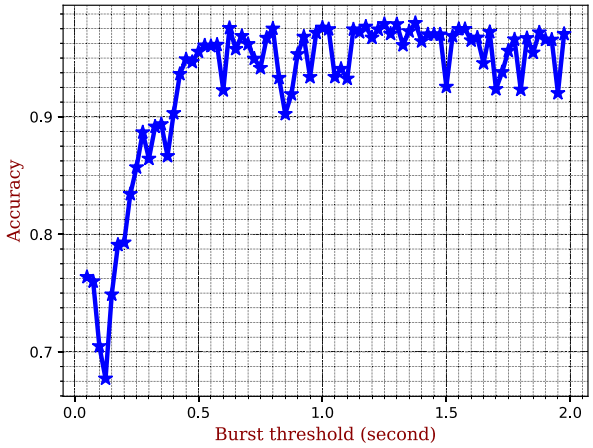


Fig. 4: Effectiveness of detection accuracy with different burst thresholds.

A. Data Description

Table II illustrates the WeChat traffic we collected using the approach described in Section III-A. We had nearly an hour worth of network traffic associated with both red packet and fund transfer transactions, half an hour worth of network traffic associated with sending / receiving of texts and pictures. During traffic collection, Wireshark was used. Half of the traffic data were used as the training set, and the remaining half for the testing set.

TABLE II: WeChat traffic data

Type of action	Duration	Total number of packets
Texts	0.5 hour	1.60k
Pictures	0.5 hour	7.80k
Red packets	1.0 hour	12.4k
Fund transfers	1.0 hour	22.8k

B. Evaluation Metrics

We evaluated the performance of the classification using the following metrics:

- Accuracy is defined as the percentage of correctly classified instance among the total number of samples. It is formulated by:

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

  where True Positive is denoted by TP, True Negative is denoted by TN, False Positive is denoted by FP, and False Negative is denoted by FN.
- Recall is the ratio of the number of TP to FP plus TP, it is given by:

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

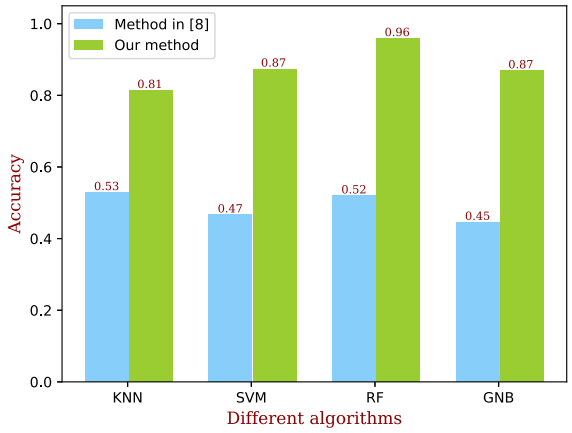


Fig. 5: Performance comparison between our designed classifier and that of [8] using different algorithms.

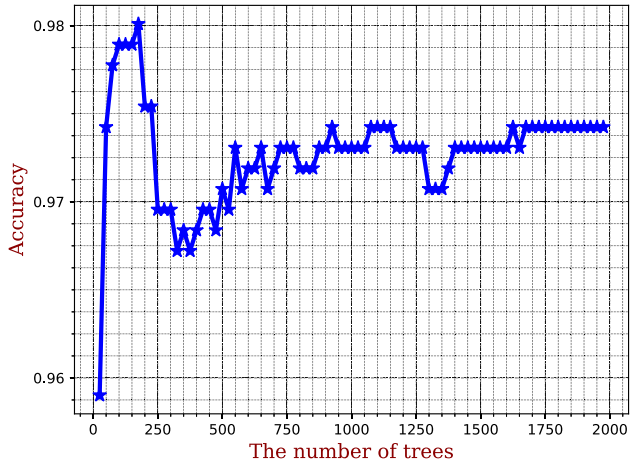


Fig. 6: Performance comparison using different RF parameters.

- F1-score considers both precision and recall, and it is formulated by:

$$F1\text{-}score = \frac{Precision \times Recall}{2 \times Precision + Recall}, \tag{3}$$

where precision is calculated by:

$$Precision = \frac{TP}{TP + FP}. \tag{4}$$

C. Selection of Burst Threshold

Before considering the classification of different user actions, we will describe how to choose the optimal burst threshold. In this section, we empirically select a reasonable value of burst threshold. As shown in Fig. 4, the accuracy achieved is over 90% when the burst threshold is more than 0.35 second. The highest accuracy is achieved around 97% when the burst threshold is around 1.25 seconds. However, the accuracy is slightly reduced when the threshold exceeds 1.5 seconds. Therefore, we empirically select 1.25 seconds as our optimal burst threshold.

D. Classification Performance

Now we evaluated the performance of our proposed approach in identifying the WeChat red packet and fund transfer transactions.

Fig. 5 illustrates the performance comparison between our proposed approach and the approach presented in [8] using four candidate classification algorithms, namely: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), RF, and Gaussian Naive Bayes (GNB). For these machine learning algorithms, we set the number of clusters as 7 in KNN, and the number of decision trees as 10 in RF. According to the findings, it is clear that our proposed method with the RF algorithm outperforms other combination approaches, and the accuracy of our approach can be as high as 96% when the RF algorithm is used.

Furthermore, we compare the performance of RF using different parameters. The ensemble method RF is a "stronger learner" consisting of a group of "weaker learner", which corresponds to different decision trees. In practice, we can also utilize the results from several decision trees trained with sub-samples of the dataset and different portions of features, in order to achieve improved performance. Therefore, we mainly compare the classification performance of different number of decision trees. As Fig. 6 illustrates, the highest accuracy is over 98% when the number of trees is 200.

Fig. 7 illustrates a confusion matrix representing the result of the classification. The 16 labels represent 16 actions, namely: sending / receiving of texts (action-1), sending / receiving of pictures (action-2), and the 14 steps of sending / receiving of red packets and fund transfers (see Table I). The findings demonstrate that our classifier can correctly and effectively identify different user actions, including texts, pictures, and every step of the WeChat red packet and fund transfers. The average accuracy is nearly 98%.

The resultant accuracy, recall and F1-scores are presented in Fig. 8, and a review of the findings suggests that our approach is effective in identifying different actions and each step of the WeChat red packet and fund transfer transactions (i.e. all actions have accuracy, recall and F1-scores of more than 93%, and the average accuracy, recall and F1-scores is up to 97%).

Table III shows the prediction accuracy of the number of WeChat red packets and fund transfers. Each action was executed 50 times (i.e. sending / receiving of 50 red packets and sending / receiving of 50 fund transfers). Surprisingly, the accuracy of receiving red packets is 100%, and the average accuracy is nearly 96%. The findings suggest that our classifier can accurately estimate the number of WeChat red packets and

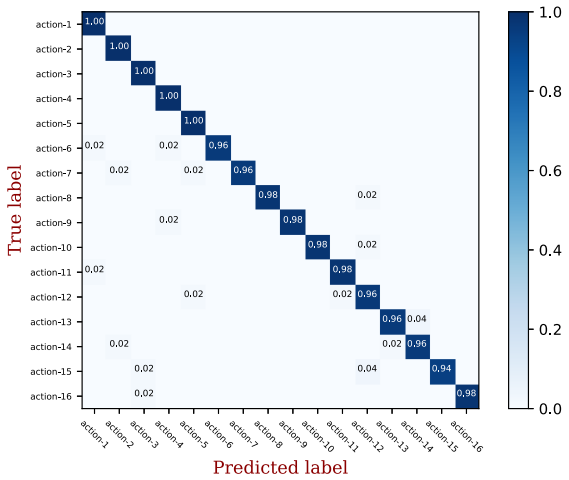


Fig. 7: Confusion matrix with different user actions.

TABLE III: Identification of the number of WeChat red packets and fund transfers

Type of actions	Number of ground truth	Predicted number	Accuracy
Sending red packets	50	48	96%
Receiving red packets	50	50	100%
Sending fund transfers	50	46	92%
Receiving fund transfers	50	48	96%
Average	/	48	96%

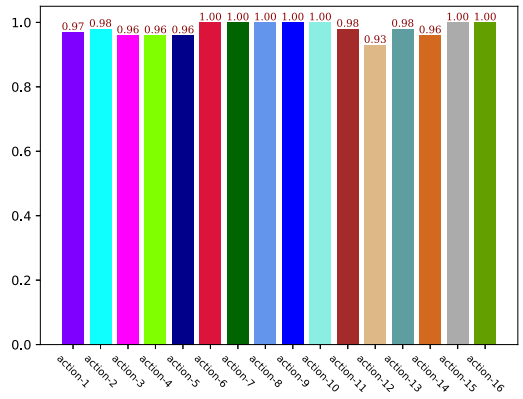
fund transfers by identifying the traffic generated by each step.

E. Efficiency

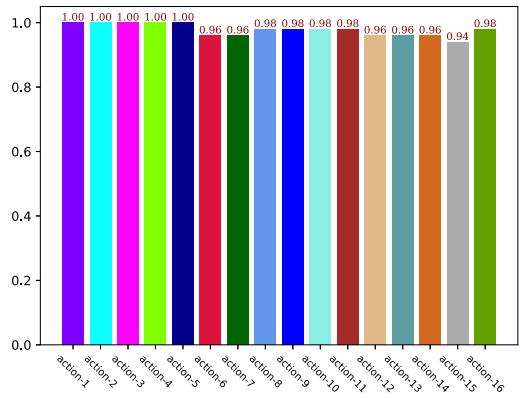
We evaluated the efficiency of our proposed method. Table IV shows the computational performance for classifying different user actions. Specifically, traffic segmentation is very time-consuming, as we need to filter out irrelevant packets, transform packets to a set of time series, and calculate all the inter-packet time.

TABLE IV: The computational performance

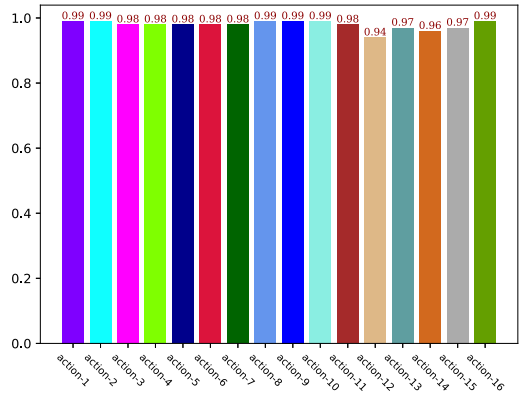
Produces	Time
Traffic segmentation (burst)	4.00 seconds
Feature extraction	2.06 seconds
Training of classification system (RF)	0.20 second
Testing of classification system (RF)	0.02 second



(a) Accuracy



(b) Recall



(c) F1-score

Fig. 8: Detection performance using proposed metrics.

V. Conclusion

WeChat is likely to remain more tightly integrated with other services, as more businesses in China adopt WeChat as a form of payment for services and merchandise. Given the potential for financial related features, such as red packet and fund transfer, to be abused for criminal purposes, it is important for WeChat service provider (Tencent) and the relevant authorities to have the capability to identify red packet and fund transfer transactions in the encrypted WeChat traffic

close to real-time.

In this paper, we presented our machine learning based approach to examine the encrypted WeChat traffic. We then demonstrated that our approach can effectively distinguish red packet and fund transfer transactions from two other popular WeChat activities (i.e. sending / receiving of texts and pictures).

Future research includes evaluating the potential of other machine learning and / or deep learning approaches that can be used to more effectively distinguish red packet and fund transfer transactions for WeChat services. In addition, future research will also include exploring the potential to collaborate with Tencent to implement our proposed approach to evaluate a larger set of data, with the aims of refining the proposed approach to be able to deal with the volume, velocity and variety of data in a real world environment.

VI. Acknowledgment

This work is funded by the cyberspace security major program in National Key Research and Development Plan of China under grant No. 2016YFB0800201, the Natural Science Foundation of China under grant No. 61702150 and No. 61572165, the State Key Program of Zhejiang Province Natural Science Foundation of China under grant No. LZ15F020003, the Key research and development plan project of Zhejiang Province under grant No. 2017C01062 and No.2017C01065.

References

[1] H. Yao, G. Ranjan, A. Tongaonkar, Y. Liao, and Z. M. Mao, "SAMPLES: Self Adaptive Mining of Persistent LExical Snippets for Classifying Mobile Application Traffic," *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking - MobiCom '15*, pp. 439–451, 2015.

[2] Q. Wang, A. Yahyavi, B. Kemme, and W. He, "I know what you did on your smartphone: Inferring app usage over encrypted data traffic," *2015 IEEE Conference on Communications and NetworkSecurity, CNS 2015*, pp. 433–441, 2015.

[3] L. Watkins, C. Corbett, and B. Salazar, "Using network traffic to remotely identify the type of applications executing on mobile devices," *... of IEEE Mobile ...*, no. 3, 2013. [Online]. Available: http://www.mostconf.org/2013/papers/21.pdf

[4] M. Conti, L. V. Mancini, R. Spolaor, and N. V. Verde, "Can't you hear me knocking: Identification of user actions on Android apps via traffic analysis," 2014. [Online]. Available: http://arxiv.org/abs/1407.7844

[5] M. Shafiq, X. Yu, A. A. Laghari, L. Yao, N. K. Karn, F. Abdesssamia, and Salahuddin, "WeChat Text and Picture Messages Service Flow Traffic Classification Using Machine Learning Technique," *Proceedings - 18th IEEE International Conference on High Performance Computing and Communications, 14th IEEE International Conference on Smart City and 2nd IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2016*, pp. 58–62, 2017.

[6] H. W. Kim and D. Choi, "Encryption is Not Enough: Inferring User Activities on KakaoTalk with Traffic Analysis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9503, pp. 254–265, 2016.

[7] S. E. Coull and K. P. Dyer, "Traffic Analysis of Encrypted Messaging Services : Apple iMessage and Beyond."

[8] Y. Fu, H. Xiong, X. Lu, J. Yang, and C. Chen, "Service Usage Classification with Encrypted Internet Traffic in Mobile Messaging Apps," *IEEE Transactions on Mobile Computing*, vol. 15, no. 11, pp. 2851–2864, 2016.

[9] W. Martin, F. Sarro, Y. Jia, Y. Zhang, and M. Harman, "A survey of app store analysis for software engineering," *IEEE Transactions on Software Engineering*, 2018.

[10] T. Watanabe, M. Akiyama, F. Kanei, E. Shioji, Y. Takata, B. Sun, Y. Ishii, T. Shibahara, T. Yagi, and T. Mori, "Understanding the origins of mobile app vulnerabilities: a large-scale measurement study of free and paid apps," *Proceedings of the 14th International Conference on Mining Software Repositories - MSR 2017*, pp. 14–24, 2017.

[11] Y. Nan, Z. Yang, M. Yang, S. Zhou, Y. Zhang, G. Gu, X. Wang, and L. Sun, "Identifying user-input privacy in mobile applications at a large scale," *IEEE Trans. Information Forensics and Security*, vol. 12, no. 3, pp. 647–661, 2017. [Online]. Available: https://doi.org/10.1109/TIFS.2016.2631949

[12] C. Ntantogian, D. Apostolopoulos, G. Marinakis, and C. Xenakis, "Evaluating the privacy of android mobile applications under forensic analysis," *Computers & Security*, vol. 42, pp. 66–76, 2014. [Online]. Available: https://doi.org/10.1016/j.cose.2014.01.004

[13] J. Farnden, B. Martini, and K.-K. R. Choo, "Privacy risks in mobile dating apps," *Proceedings of the 21st Americas Conference on Information Systems - AMCIS 2015*, pp. 1–16, 2015.

[14] R. Shetty, G. Grispos, and K. K. R. Choo, "Are you dating danger? an interdisciplinary approach to evaluating the (in)security of android dating apps," *IEEE Transactions on Sustainable Computing*, pp. 1–1, 2017.

[15] N. D. W. Cahyani, N. H. A. Rahman, W. B. Glisson, and K. R. Choo, "The role of mobile forensics in terrorism investigations involving the use of cloud storage service and communication apps," *MONET*, vol. 22, no. 2, pp. 240–254, 2017. [Online]. Available: https://doi.org/10.1007/s11036-016-0791-8

[16] C. J. D'Orazio and K.-K. R. Choo, "Circumventing ios security mechanisms for apt forensic investigations: A security taxonomy for cloud apps," *Future Generation Computer Systems*, vol. 79, pp. 247–261, 2018.

[17] Y.-Y. Teing, A. Dehghantanha, and K.-K. R. Choo, "Cloudme forensics: A case of big data forensic investigation," *Concurrency and Computation: Practice and Experience*, vol. 30(5), p. e4277, 2018.

[18] N. Malik, J. Chandramouli, P. Suresh, K. D. Fairbanks, L. Watkins, and W. H. Robinson, "Using network traffic to verify mobile device forensic artifacts," in *14th IEEE Annual Consumer Communications & Networking Conference, CCNC 2017, Las Vegas, NV, USA, January 8-11, 2017*, 2017, pp. 114–119. [Online]. Available: https://doi.org/10.1109/CCNC.2017.7983091

[19] G. Costa, A. Merlo, L. Verderame, and A. Armando, "Automatic security verification of mobile app configurations," *Future Generation Comp. Syst.*, vol. 80, pp. 519–536, 2018. [Online]. Available: https://doi.org/10.1016/j.future.2016.06.014

[20] C. Xenakis, C. Ntantogian, and O. Panos, "(u)simmonitor: A mobile application for security evaluation of cellular networks," *Computers & Security*, vol. 60, pp. 62–78, 2016. [Online]. Available: https://doi.org/10.1016/j.cose.2016.03.005

[21] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, "Robust Smartphone App Identification Via Encrypted Network Traffic Analysis," *IEEE Transactions on Information Forensics and Security*, pp. 1–13, 2017.

[22] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324