Service Usage Classification Through Encrypted Internet Traffic In Mobile Messaging Apps

Dr. T. Nirmal Raj[1], H R Rathna Bai[2]
[1, 2, 3, 4, 5] Dept of CSA
[1, 2, 3, 4, 5] MCA,Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya
(SCSVMV) University

Abstract- *The rapid adoption of mobile messaging Apps has enabled us to collect massive amount of encrypted Internet traffic of mobile messaging. The classification of this traffic into different types of in-App service usages can help for intelligent network management, such as managing network bandwidth budget and providing quality of services. Traditional approaches for classification of Internet traffic rely on packet inspection, such as parsing HTTP headers. However, messaging Apps are increasingly using secure protocols, such as HTTPS and SSL, to transmit data. This imposes significant challenges on the performances of service usage classification by packet inspection. How to exploit encrypted Internet traffic for classifying in-App usages. Specifically, we develop a system, named CUMMA, for classifying service usages of mobile messaging Apps by jointly modeling user behavioral patterns, network traffic characteristics and temporal dependencies. We first segment Internet traffic from traffic-flows into sessions with a number of dialogs in a hierarchical way. Also, we extract the discriminative features of traffic data from two perspectives: (i) packet length and (ii) time delay. CUMMA enables mobile analysts to identify service usages and analyze end-user in-App behaviors even for encrypted Internet traffic. Finally, the extensive experiments on real-world messaging data demonstrate the effectiveness and efficiency of the proposed method for service usage classification.*

Keywords- Encrypted Traffic Classification, Mobile Messaging Application, Network Traffic Analysis,Service Usage Detection, Traffic Feature Extraction.

I. INTRODUCTION

Recent years have witnessed the increased popularity of mobile messaging Apps, such as WeChat and WhatsApp. Indeed, messaging Apps have become the hubs for most activities of mobile users. For example, messaging Apps help people text each another, share photos, chat, and engage in commercial activities such as paying bills, booking tickets and shopping. Mobile companies monetize their services in messaging Apps. Therefore, service usage analytics in messaging Apps becomes critical for business, because it can help understand in-App behaviors of end users, and thus enables a variety of applications. For instance, it provides in-depth insights into end users and App performances, enhances user experiences, and increases engagement, conversions and monetization. Traditional methods for traffic classification rely on packet inspection by analyzing the TCP or UDP port numbers of an IP packet or reconstructing protocol signatures in its payload. For example, an IP packet usually has five tuples of protocol types, source address, source port, destination address and destination port. People estimate the usage types of traffic by assuming that messaging Apps consistently transmit data using the same port numbers which are visible in the TCP and UDP headers. However, there are emerging challenges for inspecting IP packet content. For example, messaging Apps are increasingly using unpredictable port numbers. Also, customers may encrypt the content of packets. In addition, governments have imposed privacy regulations which limit the ability of third parties to lawfully inspect packet contents. Moreover, many mobile apps use the Secure Sockets Layer (SSL) and its successor Transport Layer Security (TLS) as a building block for encrypted communications. To address these challenges, in this paper, we aim at developing data mining solutions for classifying encrypted Internet traffic data generated by messaging Apps into different service usage types. Note that the traffic patterns of these selected usages in WhatsApp are similar to those in WeChat. Indeed, the network traffic data of mobile messaging encode the unique patterns of both user behaviors and in-App usages and accommodation needs. It is desirable that all these needs can be satisfied without long distance traveling. Specifically, study these patterns from three perspectives: (1) behavioral structure, (2) flow characteristics, and (3) temporal dependencies. First, service usage behaviors in messaging Apps have their unique hierarchical structure. We employ the term of traffic-flow to denote the encrypted network traffic (with only time stamp and packet length information being available) generated by mobile messaging Apps, and the terms of session and dialog to represent the segments of traffic-flow in different granularity. For example, in web browsing, a session is initiated when a user opens the browser, and torn down after the browser being closed. A session usually includes multiple dialogs, each of which starts from a new tab being opened and lasts until this tab is closed. In one dialog, some users may view only one web page while others may view multiple web pages. This example shows a behavioral hierarchy. In other words, a sequence of activities can be divided into multiple sessions, and each session can be divided into multiple dialogs. Similarly, in mobile messaging Apps, a session generally starts when a user opens the App and lasts until this user closes it. Each session consists ofmultiple dialogs. Most dialogs are of single-type usage, such as text, location sharing, voice, or stream video, while other dialogs are of mixed usages.

II. IDENTIFY, RESEARCH AND COLLECT IDEA

Due to encryption in messaging apps (WhatsApp, Signal, Telegram), we can't see what users send — but we can still see traffic patterns. These patternsmay reveal what kind of service the user is using (texting, calling, sharing media).The first step in this project is to identify the key factors involved in the classification of encrypted internet traffic in mobile messaging apps. This involves understanding the structure of encrypted traffic—specifically, how metadata such as packet size, flow duration, and inter-arrival times can be used for analysis, even when the content itself is encrypted. Next, research needs to be conducted on existing methods and techniques that have been employed to classify traffic in other domains. This could involve a review of machine learning methods, deep learning approaches, and techniques such as traffic fingerprinting that attempt to classify encrypted traffic based on observable patterns rather than the actual data content. A key part of this research will be exploring challenges such as class imbalance, non-stationarity of traffic patterns, and privacy concerns that influence the success and ethical considerations of traffic classification. The project should also look into open-source datasets and research papers that have already explored these areas in mobile messaging apps, identifying potential gaps in the current body of knowledge. To collect ideas, the project should focus on gathering insights from various academic papers, industry reports, and experimental works to understand the state-of-the-art classification methods for encrypted internet traffic. This would involve examining how techniques like neural networks, decision trees, and support vector machines (SVM) have been applied to classify encrypted data. The idea is to understand the advantages and limitations of each approach in the context of mobile messaging traffic. User behavior modeling based on traffic metadata is another key concept to explore, as this can enhance the precision of classification models. Additionally, the project could investigate how anonymization techniques and encryption methods may affect traffic classification accuracy. Finally, the collecting of ideas should also include practical considerations for implementing a classification system using HTML and JavaScript for simulating and visualizing the methods in an accessible way. In summary, this phase of the project focuses on gathering the necessary research to inform the development of a robust classification system for encrypted internet traffic in mobile messaging apps.

III. WRITEDOWNYOURSTUDIESAND FINDINGS

In this project, the focus was to explore how encrypted internet traffic in mobile messaging applications can be analyzed and classified based on traffic metadata such as packet sizes, inter-arrival times, and flow durations. Our research started by identifying existing approaches in the literature, such as those utilizing machine learning algorithms like decision trees, random forests, and deep learning techniques for traffic classification. Previous studies demonstrated that while encrypted traffic is challenging to analyze due to the lack of access to the actual content, useful patterns can still be derived from metadata. The first finding of this study confirmed that packet size and inter-arrival times were the most important features for distinguishing between different types of mobile messaging traffic, such as text messages, voice calls, and media transfers. As part of the methodology, we simulated encrypted traffic from various messaging services and used machine learning models to classify the data. Models such as RandomForests and Support Vector Machines (SVM) were particularly effective in handling the class imbalance often found in traffic datasets, where voice and video calls are much less frequent compared to text messages. However, despite the promising results, several challenges were encountered, including the non-stationary nature of traffic patterns, meaning that user behavior could change over time, complicating the classification process. Furthermore, privacy concerns related to the traffic analysis were noted, particularly when dealing with real-world datasets. The findings of this research suggest that while classification of encrypted traffic is feasible using traffic features, further work is needed to address privacy issues and improve model robustness through advanced techniques such as deep learning. Future directions include exploring anomaly detection and adaptive models to enhance the accuracy and reliability of traffic classification systems.

IV. GETPEERREVIEWED

Throughout this project, a comprehensive review of peer-reviewed literature was conducted to understand the state-of-the-art methods in encrypted traffic analysis and classification. Several key papers highlighted the challenges and solutions in this field. For instance, Conti et al. (2015)

[1] examined how encrypted network traffic in Android apps could be analyzed without decrypting the content, using traffic features such as packet size and flow durations. This research laid the foundation for the classification approach in our project. Peer-reviewed works like those by Lotfollahi et al. (2020)

[2] further validated the use of deep learning techniques to analyze encrypted traffic, demonstrating that advanced neural networks can effectively classify encrypted data by identifying patterns in metadata. Additionally, Anderson and McGrew (2017)

[3] discussed the importance of accounting for noisy labels and non-stationary behavior in traffic data, reinforcing the need for adaptable models in real-world applications. These insights, derived from peer-reviewed sources, significantly shaped the methodology of this project, confirming the effectiveness of machine learning models such as Random Forests and Support Vector Machines (SVM) for handling encrypted traffic classification. Moreover, privacy concerns noted in studies like Apthorpe et al. (2017)

[4] have been critical in shaping the ethical considerations of the project. The peer-reviewed literature provided not only validation for the techniques used but also identified gaps and challenges that continue to drive future research in the field. This collaborative understanding from scholarly research helped refine the project's approach and solidified its academic grounding.

V. IMPROVEMENT AS PER REVIEWER COMMENTS

.

Granularity of Traffic Features:

- Reviewer feedback suggested incorporating additional traffic patterns to enhance classification accuracy.
- **Improvement**: Added features like **flow duration** and **directionality** to better differentiate between service types (e.g., text messaging, voice calls, media transfers).

Handling Class Imbalance:

- Reviewers pointed out the imbalance in traffic data, where certain types of services (e.g., voice/video calls) are less frequent.
- **Improvement**: Implemented **resampling techniques** and explored **cost-sensitive learning** methods to give more weight to underrepresented classes, improving classification for less frequent services.

Privacy Concerns:

- Reviewers raised concerns about the ethical implications of traffic analysis and the use of real-world data.
- **Improvement**: Integrated **anonymization techniques** and added safeguards to ensure ethical data usage and privacy protection during analysis.

Evaluation Metrics:

- The original evaluation relied solely on **accuracy**, which can be misleading in imbalanced datasets.
- **Improvement**: Expanded evaluation metrics to include **precision**, **recall**, and **F1-score** to give a more comprehensive understanding of model performance, especially in cases of class imbalance.

Advanced Deep Learning Models:

- Reviewers recommended exploring more sophisticated deep learning techniques for improved classification accuracy
- **Improvement**: Tested **Convolutional Neural Networks (CNNs)**, which were found to enhance model performance in identifying complex patterns in encrypted traffic.

VI. CONCLUSION

We developed a system for classifying service usages using encrypted Internet traffic in mobile messaging Apps by jointly modelling, behavior structure, network traffic characteristics, and temporal dependencies. There are four modules in our system including traffic segmentation, traffic feature extraction, service usage prediction, and outlier detection and handling. Specifically, we first built a data collection platform to collect the traffic-flows of in-App usages and the corresponding usage types reported by mobile users. We then hierarchically segment these traffic from traffic-flows to sessions to dialogs where each is assumed to be of individual usage or mixed usages.

APPENDIX

The Appendix section of this project provides supplementary information that supports the research and analysis. It includes the detailed data collection process, additional experiments, and code snippets used throughout the study.

A. SYSTEM CONFIGURATION

HARDWARE:

- System : Pentium IV 3.5 GHz or Latest Version
- Hard Disk : 40 GB
- Monitor : 14’ Colour Monitor
- Mouse : Optical Mouse
- Ram : 1 GB

SOFTWARE:

- Operating system: Windows XP or Windows 7, Windows 8
- Frontend :Html ,Javascript.
- Backend : JSP (java server pages)
- Data Base: My SQL Server
- Documentation : MS Office
- IDE : Eclipse Galileo
- Development Kit : JDK 1.6
- Server : Tomcat 6.0

B FUNCTIONAL MODULES OVERVIEW

Traffic Feature Extraction Module:

- Extracts essential metadata features from the encrypted traffic, such as packet size, inter-arrival times, flow duration, and flow direction.
- This module processes raw network traffic data and converts it into usable features for classification**.**

Data Collection Module:

- Collects both simulated and real-world encrypted traffic samples.
- Simulated traffic data is generated based on various messaging services (e.g., text, voice, media transfers) using a network traffic simulator.
- Real-world traffic data is collected using tools like Wireshark, capturing packet-level details for further analysis.

Traffic Classification Module:

- Implements machine learning algorithms (e.g., Random Forests, Support Vector Machines (SVM)) to classify encrypted traffic into different service categories.
- The classification model is trained using extracted features and evaluated based on accuracy and other performance metrics**.**

VII. ACKNOWLEDGMENT

I would like to express my sincere gratitude to my project guide**, Dr. T. Nirmal Raj**, for his invaluable guidance, constant encouragement, and dedicated support throughout the duration of this research work. His expertise and constructive feedback were instrumental in shaping the direction and outcome of this project.

I also extend my thanks to the faculty members of the Department of Information Technology for providing the necessary facilities and a conductive environment for research. Lastly, I thank my peers and family for their continuous motivation and support during the course of this project.

REFERENCES

[1] **S. Urman**, *Oracle PL/SQL Programming*. McGraw-Hill, 1996.

[2] **J. R. Groff** and **P. N. Weinberg**, *SQL: The Complete Reference*, 3rd ed. McGraw-Hill, 2009.

[3] **H. Schildt**, *Java: The Complete Reference*, 5th ed. McGraw-Hill, 2002.

[4] **L. Vanhelsuwe**, **A. Yee**, and **I. Phillips**, *Mastering Java*. Sybex Inc., 1996.

[5] **Y. Shiran** and **T. Shiran**, *Learn Advanced JavaScript Programming*. Wordware Publishing, 1997.

[6] **M. Pistoia** and **M. Nagnur**, *Java 2 Network Security*, 2nd ed. Prentice Hall, 1999.

[7] **S. Oaks**, *Java Security*, 2nd ed. O'Reilly Media, 2001.

[8] **S. Siddiqui** and **P. Jain**, *J2EE Professional Projects*. Premier Press, 2002.

[9] **N. Todd** and **M. Szolkowski**, *JavaServer Pages: Developer's Handbook*. Sams Publishing, 2003.

[10] **S. Holzner**, *HTML Black Book: The Programmer's Complete HTML Reference*. Paraglyph Press, 2000.

[11] **P. Patel** and **K. Moss**, *Java Database Programming with JDBC*. Coriolis Group, 1997.

[12] **R. S. Pressman** and **B. Maxim**, *Software Engineering: A Practitioner's Approach*, 9th ed. McGraw-Hill Education, 2020.