

2020

# USER PROFILING BASED ON NETWORK APPLICATION TRAFFIC MONITORING

Shaman, Faisal

<http://hdl.handle.net/10026.1/16440>

---

<http://dx.doi.org/10.24382/652>

University of Plymouth

---

*All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.*

## **Copyright Statement**

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the author's prior consent.



**UNIVERSITY OF  
PLYMOUTH**

**USER PROFILING BASED ON NETWORK  
APPLICATION TRAFFIC MONITORING**

by

**FAISAL SHAMAN**

A thesis submitted to the University of Plymouth  
in partial fulfilment for the degree of

**DOCTOR OF PHILOSOPHY**

School of Engineering, Computing and Mathematics

**July 2020**

## **Acknowledgements**

First and foremost, all praise and gratitude are due to Allah Almighty, the All Merciful, for helping me and facilitating me in tackling all the challenges throughout this PhD that could otherwise have constrained my study.

I am deeply indebted, and my most sincere thanks and appreciation go to my beloved parents, for their considerable help and support, kindness, abundant love, and prayers for my study, and I ask Allah to reward them with the best. I would also like to take this opportunity to express my sincere gratitude to my brothers and sisters for their immeasurable love and encouragement through this important stage of my life. Special thanks go to my brother, Mouhammed Shaman, for his kindness and help during my PhD journey.

I also owe many thanks to my wife, Amal, and to my children, Turki, Nawaf and Rakan, for their patience, endless support, and incredible care in assisting me throughout this endeavour. They all stood alongside me and provided me with an abundance of love and support, even when spending days, nights, and holidays without me. I really appreciate your endless support and help through this PhD journey. I am eternally grateful.

I would, of course, like to extend my most sincere thanks and my heartfelt appreciation to my supervision team, Dr Bogdan Ghita and Professor Nathan Clarke, for their guidance, support, wisdom, help and a sympathetic ear. Their experience and professionalism have been invaluable throughout my PhD journey and without their valuable comments and advice, I would not have been able to make this a success, so thank you.

I would also like to express my thanks to my research colleagues at the Centre for Security, Communication and Network Research, who have been my motivation and inspiration and with whom I have held interesting discussions during this PhD journey.

I would also like to thank all my colleagues at the University of Tabouk in the Kingdom of Saudi Arabia, for allowing me to take this great opportunity to complete my PhD degree and for their support and assistance.

Last, but not least, many thanks and much appreciation must go to my Government of The Custodian of the Two Holy Mosques for sponsoring my undertaking of the research project and for their generous support and valuable comments.

I dedicate this to my wife and children: Amal, Turki, Nawaf and Rakan.

## Author's Declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award without prior agreement of the Doctoral College Quality Sub-Committee.

Work submitted for this research degree at the University of Plymouth has not formed part of any other degree either at the University of Plymouth or at another establishment.

This study was financed with the aid of a scholarship from the Kingdom of Saudi Arabia - Royal Embassy of Saudi Arabia Cultural Bureau in London.

### Publications:

- Shaman, Faisal, Bogdan Ghita, Nathan Clarke, and Abdulrahman Alruban. "User Profiling Based on Application-Level Using Network Metadata." In *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*, pp. 1-8. IEEE, 2019.
- F. Shaman, F., Ghita, B., Clarke, N. and Alruban, A. "User Profiling Using Application-Level Sessions Timing Resolution." In *13th International Conference for Internet Technology and Secured Transactions (ICITST-2018)*, 2018, pp. 46-52.

Word count of main body of thesis: 48,241

Signed Faisal Shaman

Date 14 September 2020

# **Abstract**

## **User Profiling Based on Network Application Traffic Monitoring**

**Faisal Shaman**

There is increasing interest in identifying users and behaviour profiling from network traffic metadata for traffic engineering and security monitoring. However, user identification and behaviour profiling in real-time network management remains a challenge, as the activities and underlying interactions of network applications are constantly changing. User behaviour is also changing and adapting in parallel, due to changes in the online interaction environment. A major challenge is how to detect user activity among generic network traffic in terms of identifying the user and his/her changing behaviour over time. Another issue is that relying only on computer network information (Internet Protocol [IP] addresses) directly to identify individuals who generate such traffic is not reliable due to user mobility and IP mobility (resulting from the widespread use of the Dynamic Host Configuration Protocol [DHCP]) within a network. In this context, this project aims to identify and extract a set of features that may be adequate for use in identifying users based on their network application activity and timing resolution to describe user behaviour. The project also provides a procedure for traffic capturing and analysis to extract the required profiling parameters; the procedure includes capturing flow traffic and then performing statistical analysis to extract the required features. This will help network administrators and internet service providers to create user behaviour traffic profiles in order to make informed decisions about policing and traffic management and investigate various network security perspectives.

The thesis explores the feasibility of user identification and behaviour profiling in order to be able to identify users independently of their IP address. In order to maintain privacy and overcome the issues associated with encryption (which exists on an increasing volume of network traffic), the proposed approach utilises data derived from generic flow network traffic (NetFlow information). A number of methods and techniques have been proposed in prior research for user identification and behaviour profiling from network traffic information, such as port-based monitoring and profiling, deep packet inspection (DPI) and statistical methods. However, the statistical methods proposed in this thesis are based on extracting relevant features from network traffic metadata, which are utilised by the research community to overcome the limitations that occur with port-based and DPI techniques. This research proposes a set of novel statistical timing features extracted by

considering application-level flow sessions identified through Domain Name System (DNS) filtering criteria and timing resolution bins: one-hour time bins (0-23) and quarter-hour time bins (0-95). The novel time bin features are utilised to identify users by representing their 24-hour daily activities by analysing the application-level network traffic based on an automated technique. The raw network traffic is analysed based on the development of a features extraction process in terms of representing each user's daily usage through a combination of timing features, including the flow session, timing and DNS filtering for the top 11 applications. In addition, media access control (MAC) and IP source mapping (in a truth table) is utilised to ensure that profiling is allocated to the correct host, even if the IP addresses change.

The feature extraction process developed for this thesis focuses more on the user, rather than machine-to-machine traffic, and the research has sought to use this information to determine whether a behavioural profile could be developed to enable the identification of users. Network traffic was collected and processed using the aforementioned feature extraction process for 23 users for a period of 60 days (8 May-8 July 2018). The traffic was captured from the Centre for Cyber Security, Communications and Network Research (CSCAN) at the University of Plymouth.

The results of identifying and profiling users from extracted timing features behaviour show that the system is capable of identifying users with an average true positive identification rate (TPIR) based on hourly time bin features for the whole population of ~86% and ~91% for individual users. Furthermore, the results show that the system has the ability to identify users based on quarter-hour time bin features, with an average TPIR of ~94% for the whole population and ~96% for the individual user.

# Table of Contents

1. Introduction .....	2
1.1 Introduction.....	2
1.2 Traffic classification .....	3
1.3 User behaviour traffic profiling .....	4
1.3.1 Growth of internet traffic .....	5
1.4 Research aim and objectives.....	8
1.5 Thesis structure.....	9
2 User and application traffic classification .....	13
2.1 Introduction.....	13
2.2 User profiling.....	14
2.3 Traffic collection and processing.....	15
2.3.1 Network monitoring tools .....	18
2.4 Analysis techniques .....	21
2.4.1 Deep packet inspection .....	21
2.4.2 Machine learning analysis.....	22
2.5 Previous work on user profiling based on network application traffic.....	27
2.5.1 Network traffic classification techniques.....	27
2.5.2 Port-based techniques .....	28
2.5.3 Payload-based techniques .....	30
2.5.4 Statistical-based techniques .....	31
2.5.5 Machine learning.....	34

2.6 Literature on behavioural profiling.....	47
2.6.1 User identification.....	47
2.6.2 Intrusion detection systems.....	50
2.7 Discussion.....	53
2.8 Conclusion.....	56
3 User behaviour profiling using an application-level flow sessions.....	59
3.1 Introduction.....	59
3.2 Flow inter-arrival timing, sessions and features.....	60
3.2.1 Flow session definition and traffic splitting.....	62
3.2.2 Flow session timing resolution features.....	65
3.2.3 Flow session extracted statistical features.....	72
3.3 Discussion.....	75
3.3.1 Research questions identified.....	75
3.4 Conclusion.....	76
4 Methodology and data collection.....	79
4.1 Introduction.....	79
4.2 General block diagram for the proposed system.....	80
4.3 Data collection.....	82
4.4 Data pre-processing.....	83
4.4.1 Traffic dump.....	84
4.4.2 IP source and MAC mapping.....	84
4.4.3 Domain name system lookup.....	86

4.5 Feature extraction process .....	90
4.6 Gradient boosting classification .....	92
4.7 Discussion.....	94
4.8 Conclusion .....	95
5 Evaluation and analysis .....	98
5.1 Introduction.....	98
5.2 User identification and profiling using application-level session hourly timing features.....	99
5.2.1 User identification rates: experimental results.....	100
5.2.2 Individual user TPIR ranks .....	104
5.2.3 Confusion matrix.....	107
5.2.4 Feature importance.....	111
5.2.5 Descriptive analysis of mean and standard deviation variance.....	114
5.3 A novel feature set for user identification and profiling using application-level quarter-hour session timing features.....	122
5.3.1 User identification rates: experimental results.....	124
5.3.2 Individual user TPIR ranks .....	127
5.3.3 Confusion matrix.....	130
5.3.4 Feature importance.....	134
5.3.5 Descriptive analysis of mean and standard deviation variance.....	138
5.4 Discussion.....	145
5.5 Conclusion .....	148
6 Overall architecture .....	150

6.1 Introduction.....	150
6.2 The general architecture of user behaviour profiling using an application-level flow sessions .....	151
6.2.1 Data Collection Engine .....	152
6.2.2 Traffic Pre-Processing Engine .....	153
6.2.3 Behaviour Classification Engine.....	155
6.2.4 Security Decision Engine.....	157
6.3 Case study of user behaviour profiling using application-level flow sessions ..	159
6.4 Conclusion .....	161
7 Conclusions and future work.....	164
7.1 Achievements of the research .....	164
7.2 Limitations of the research .....	165
7.3 Suggestions and future research scope .....	167
7.4 The future of research into user identification and behaviour profiling .....	168
References .....	169
Appendix .....	177
Ethical approval documentation .....	199

## List of Figures

Figure 1.1: Internet traffic growth from 2017 to 2022 (cisco global, 2019).....	6
Figure 1.2: Internet data traffic growth in 2019 (Internet world stat, 2019).....	7
Figure 1.3: Global mobile data traffic, 2017 to 2022 (Cisco VNI Mobile, 2019).....	7
Figure 2.1: NetFlow (nfdump) architecture (Haag, 2006).....	20
Figure 2.2: Neural network structure and weights.....	25
Figure 3.1: Distribution of the hourly flow inter-arrival time threshold.....	69
Figure 3.2: Distribution of the flow session extracted statistical features.....	74
Figure 4.1: Framework of proposed user behaviour profiling using an application-level flow sessions.....	82
Figure 4.2: Number of flows for each user for the observed data.....	86
Figure 4.3: Filtered applications based on DNS.....	89
Figure 4.4: Number of samples for each user.....	92
Figure 5.1: Hourly timing features – individual users’ TPIR.....	106
Figure 5.2: Hourly timing features for individual users (recall, precision and F1 score) .....	111
Figure 5.3: Hourly timing features scores.....	114
Figure 5.4: Hourly level of top five mean and SD variance.....	119
Figure 5.5: Users’ TPIR ranks for quarter-hour timing features.....	129
Figure 5.6: Quarter-hour for individual users (recall, precision and F1 score).....	134
Figure 5.7: Quarter-hour feature importance scores.....	138
Figure 5.8: Quarter-hour top five features mean and SD variance.....	144
Figure 5.9: Comparison of results for the flow session timing features.....	146
Figure 5.10: Individual user TPIR comparison.....	147
Figure 6.1: Overall architecture for user behaviour profiling using an application-level flow sessions.....	151
Figure 6.2: Data collection engine.....	152
Figure 6.3: Traffic pre-processing engine.....	153
Figure 6.4: Proposed timing feature.....	156
Figure 6.5: Behaviour classification engine.....	157
Figure 6.6: Security decision engine.....	159

## List of Tables

Table 2.1: Basic features and derivative features in flow records (Jiang et al., 2007) ...	36
Table 3.1: Hourly timing resolution.....	68
Table 3.2: Quarter-hour timing resolution .....	72
Table 3.3: Flow session timing activity features.....	73
Table 4.1: MAC address and IP source mapping.....	85
Table 4.2: Features extracted after the domain name lookup process .....	87
Table 5.1: Hourly timing features - users' traffic identification rate results.....	102
Table 5.2: Hourly timing features – users' TPIR ranks (features set 2) .....	105
Table 5.3: Hourly timing features confusion matrix (features set 2) .....	109
Table 5.4: Hourly timing features importance weights.....	113
Table 5.5: Hourly timing features - top five mean and SD variances.....	115
Table 5.6: Quarter-hour features – users' traffic identification rate results .....	126
Table 5.7: Users' TPIR ranks for set 5 using quarter-hour timing features .....	128
Table 5.8: Quarter-hour timing features confusion matrix (features set 5).....	131
Table 5.9: Quarter-hour timing feature importance weights.....	137
Table 5.10: Quarter-hour top five feature means and SD variances .....	138
Table 6.1: DNS engine storage .....	154

## Glossary

ACK	-	Acknowledgement
ARP	-	Address Resolution Protocol
bpp	-	Bytes per packet
bps	-	Bits per second
CSCAN	-	Centre for Security Communications and Network Research
d2s	-	Destination to source
DHCP	-	Dynamic Host Configuration Protocol
DNS	-	Domain Name System
DPI	-	Deep packet inspection
dst	-	Destination
FIN	-	Final
FN	-	False negative
FP	-	False positive
FTP	-	File Transfer Protocol
IANA	-	Internet Assigned Numbers Authority
IDS	-	Intrusion detection system
IP	-	Internet Protocol
ISP	-	Internet service provider
MAC	-	Media access control
ML	-	Machine learning
mps	-	Mean packet size
NIC	-	Network interface card
pps	-	Packets per second
PSH	-	Push
RST	-	Reset
s2d	-	Source to destination

src	-	Source
SVM	-	Super vector machine
SYN	-	Synchronise
TCP	-	Transmission Control Protocol
TN	-	True negative
ToS	-	Type of service
TP	-	True positive
TPIR	-	True positive identification rate
UDP	-	User Datagram Protocol



# **Chapter One**

## **Introduction**

# 1. Introduction

## 1.1 Introduction

User identification and behaviour profiling from generic network traffic metadata have become critical parts of network and traffic management due to the massive use of computer systems and applications, as well as their increased complexity (Bakhshi and Ghita, 2015). It has also become necessary to be able to identify security breaches and enforce organisational policy, as well as to provide intelligent routing decisions to help network administrators and security investigators with infrastructure traffic monitoring. User profiling based on features extracted from network traffic metadata (source to destination [s2d] packet size and inter-arrival time) encourages the internet service provider (ISP) to know the user and how that individual has interacted with the system, in order to enhance the security and policies of the particular organisation. The translation of each user activity involves a network footprint of user interaction that can be implemented by employing a method of extracting relevant features for user identification and behaviour profiling (Oliveira, 2011). Investigating network traffic metadata to identify and profile users is a challenging task in this research area, as the nature of online applications and interaction changes over time, while user behaviour constant component has slight variance. In addition, an Internet Protocol (IP)-agnostic solution allows for a reduction in the cross-layer monitoring of users, although users can indeed be linked through their authentication profiles with the IP addresses they have been allocated (Dehghani *et al.*, 2010). Therefore, a need exists for an approach that is able to classify and identify the network traffic associated with an individual in order to produce user behaviour traffic profiling that can enhance the ability of a security manager or ISP to address security issues that affect the organisation.

Internet services have become part of our daily lives and the majority of tasks undertaken are now based on applications, such as managing the booking of airline flights, acquiring

utility services (e.g., water and electricity) and conducting online banking transactions. Moreover, using these applications and internet services can enhance our ability to deal with everyday activities in an easy and quick way, which further increases the need to manage the growth in network traffic.

## **1.2 Traffic classification**

Traffic classification and analysis is utilised in traffic engineering to provide efficient management of the networked world. Classifying users from network flows and packet capture has been attempted in several previous research studies, which have proposed various alternative classification techniques. Network security and management systems need users' traffic identification and classification to be extracted at the application level, as this should lead to a user behaviour profile that could enhance policy and help manage the organisation (Alcock and Nelson, 2012). An individual user behaviour profile can be generated from network traffic, which is then utilised to identify a user based on his/her previous interactions with the application during a specific time slot. The examination of users' interactions and monitoring applications can create a user template that can be used to decide whether this kind of activity belongs to a legitimate user (Heer *et al*, 2001).

As demand for using networks and applications over the internet increases, traffic classification becomes an increasingly critical issue (Rossi & Valenti 2010; Heer & Chi n.d. 2004; Zhang et al. 2012; No & Jamuna 2013). Network security and management require traffic classification to differentiate between users in terms of identifying their different behaviour profiles. Network administrators can use these behaviour profiles to consider how decisions about configuration changes will affect the rest of the components in their network. A user's behaviour profile can also be used by the security administrator to identify behaviours that violate policy (Balram and Wilscy, 2014).

A number of research studies have focused on providing traffic classification using port-based techniques, which handle classification through well-known port numbers (Nguyen & Armitage 2008; Dainotti et al. 2012). However, the effectiveness of this approach has disappeared because random port numbers can be used for different applications. To address this problem, payload-based techniques have been proposed to compare packet payload information in order to address the randomness of port numbers (Finsterbusch et al. 2014; Shaikh et al. n.d.). However, the problem with the payload-based technique is encryption, which prevents the analyser from going deeply into the content of the packet. To overcome such limitations, statistical techniques have been proposed to identify users without accessing the payload information. Novel user identification and a traffic classification methodology are then necessary for accurate traffic classification and user identification based on statistical methods and a novel features extraction process. Additional research on the flow-based technique and statistical analysis is also required to extract novel features, which will enhance the ability of a system to identify users by investigating their behaviour and activity based on application-level traffic analysis and classification.

### **1.3 User behaviour traffic profiling**

User behaviour traffic profiling represents an area of significant interest for the networking community, as user profiling and statistical analysis are critical steps for workload characterisation, capacity planning and network policy in computer networks (Kihl and Odling, 2010). Hence, a deep investigation of a user profile based on the user's activity will enhance the security and management of the network by helping the administrator to take important decisions based on the user's profile information. Furthermore, research has become more focused on traffic profiling based on overall traffic levels, the distribution of traffic on specific ports and application layer profiling, coupled with time-based mapping and statistical analysis (Xu, Zhang and Bhattacharyya,

2008). Prior research has suggested approaches to user traffic classification, such as flow- and host-based techniques, as well as clustering, to build common behaviour modules for traffic profiling; however, existing methods do not aggregate user behaviour and application profiling (Yang *et al*, 2011).

Clustering applications to extract user traffic profiling from the application used may provide detailed information for traffic patterns, but applying it to real-time network topology for network management through characterisation is more challenging. Following an initial characterisation, user behaviour and activity may change over time and, with it, the defining profiles may need to be adjusted. Using a combination of throughput, flow statistics and timing information may prove sufficient to produce a reliable user profile pattern for identifying individual users (Yang *et al*, 2015).

### **1.3.1 Growth of internet traffic**

As shown in Figure 1.1, internet traffic is increasing dramatically, from 122 exabytes per month in 2017 to an expected 396 exabytes per month by the end of the period shown. The first three years show significant growth but a boom in internet traffic is expected from 2020 onwards, when it will increase by many thousands of times. Therefore, traffic classification engineering has become more challenging due to the volume of traffic transmitted, which requires more space for storage and a complicated algorithm to classify the traffic. By 2022, different types of smartphone and tablet, as well as new technologies, are expected, such as the internet of things and more complicated applications, which will cause users to migrate from using a PC at home or at work to using smartphones anywhere, which will further increase the amount of traffic transmitted over the internet.

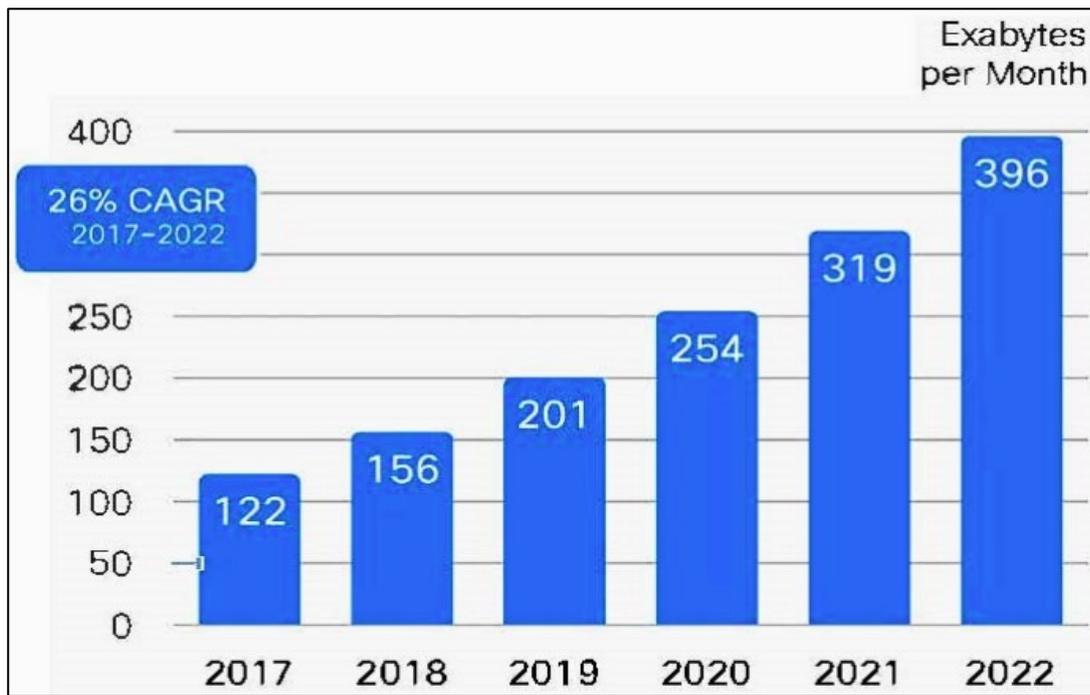


Figure 1.1: Internet traffic growth from 2017 to 2022 (cisco global, 2019)

In addition, there is a significant trend expected in technology companies, such as Amazon and Google, releasing and using online applications. The usage of applications is expected to increase 750 times between 2020 to 2022, which will increase the growth of internet traffic to 396 exabytes per month (Cisco global, 2019).

In 2019, internet data traffic increased to 49.6%, with the percentage growth varying from region to region. As shown in Figure 1.2, North America saw the highest growth percentage, at 89.4%, compared with 87.7% for the next largest region - Europe. The data traffic in Latin America was 68.9%, with Oceania/Australia close behind at 68.9%. Data traffic in the Middle East reached 67.9% and 54.2% in Asia, compared with 39.6% in Africa (Internet World Stats, 2019). Therefore, this huge increase in traffic has led to far greater importance in terms of managing traffic volumes to enhance the security in the network environment.

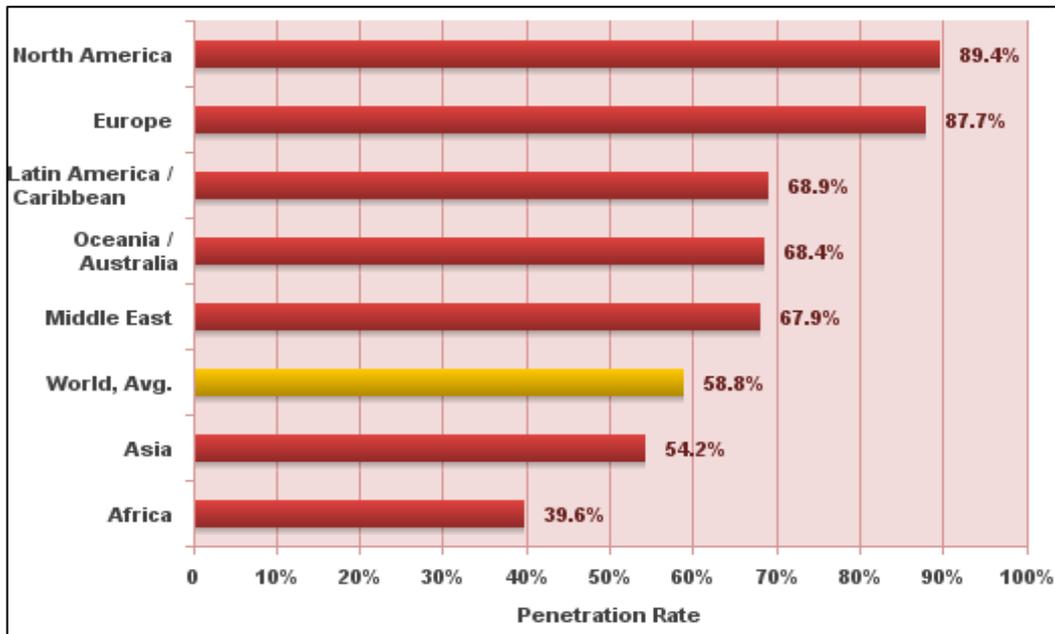


Figure 1.2: Internet data traffic growth in 2019 (Internet world stat, 2019)

Monthly growth in mobile data traffic from 2017 to 2022 is shown in Figure 1.3. Mobile data traffic was at the level of 12 exabytes per month in 2017, the growth rate increasing to 19 exabytes in 2018, with an expected monthly data growth to 29 exabytes in 2019 and 41 exabytes in 2020. This growth is expected to continue for the near future, to reach 57 exabytes in 2021 and then 77 exabytes in 2022 (*Internet world stats, 2019*).

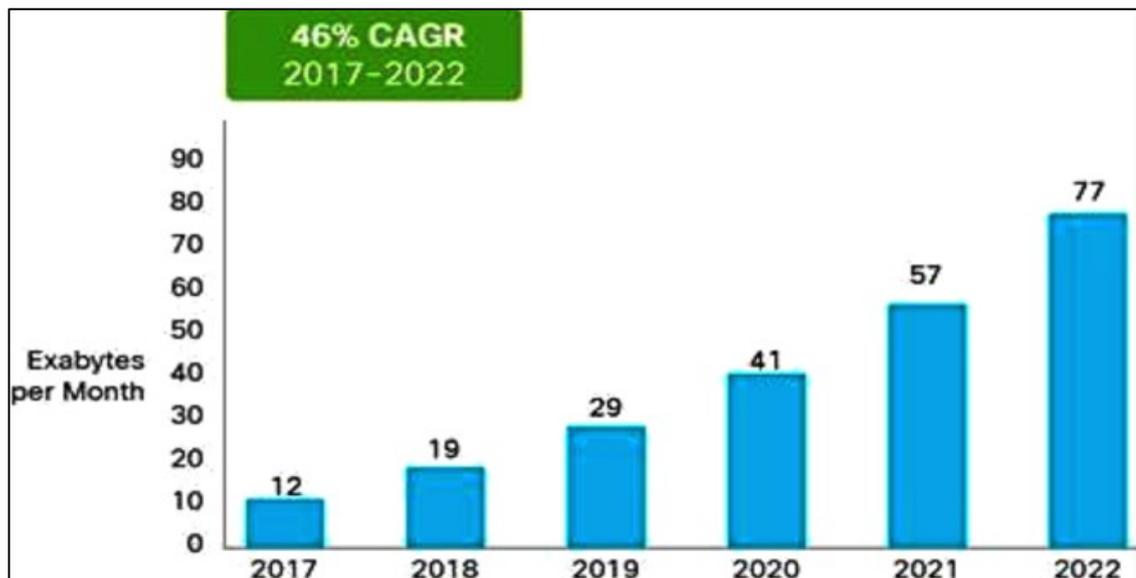


Figure 1.3: Global mobile data traffic, 2017 to 2022 (Cisco VNI Mobile, 2019)

As a result of a significant rise in the usage of computer systems and applications, as well as their increased complexity, classifying applications and user behaviour has become critical for network managers and ISPs to identify security issues and/or malicious activity happening within the network. It is, therefore, necessary to create traffic profiles that can evaluate and detect abnormal network traffic by comparing it against a ‘normal’ traffic profile (*Cisco VNI Mobile*, 2019). Using traffic profiling information in the area of computer security can help significantly in the detection of rogue users, malicious activity, policy violations and unauthorised data exfiltration, and will, in many cases, prevent them from happening (Robinson et al., 2006). To address the huge growth in internet traffic, researchers have proposed various types of traffic classification and identification techniques. As user identification and profiling has become increasingly important for information security and traffic management, identifying users’ activity and profiles will enhance the security of the organisation’s networks. However, with the growth of internet traffic, as well as the widespread use of the Dynamic Host Configuration Protocol (DHCP), IP addresses are subject to frequent change. As such, a need exists for an approach that can identify individuals through the footprint of network traffic’s novel features, rather than IP addresses, and is based on an automated and reliable process. This research project builds on this need to monitor and manage generic network traffic in terms of user identification and behaviour profiling, based on an application-level session timing resolution to address some of the limitations that have occurred because of the growth in network traffic data around the world, as well as the frequent changes in user behaviour.

#### **1.4 Research aim and objectives**

The main aim of this research was to design, develop and evaluate an efficient mechanism for profiling computer network users based on the network traffic metadata generated (i.e.,

network flow information). To achieve this, the following research objectives were established:

- To investigate the current state-of-the-art research on user traffic profiling.
- To extract a set of features that could be used for identifying patterns in network application activities and describing user behaviour; the features are likely to be based on traffic information analysis.
- To collect computer network traffic from real users' interactions to be used in the evaluation phase.
- To design and develop a data processing pipeline for identifying network users' profiling based on the features identified.
- To evaluate and analyse how identifiable and unique user behaviour profiling is over time in the context of the dynamic, changing nature of network applications. This includes investigating the effect of different time resolutions (i.e., hours and quarter hours) when processing the collected data.
- To propose a system architecture for user profiling based on a network application traffic monitoring system.

## **1.5 Thesis structure**

The rest of the thesis is structured as follows. Chapter Two provides background information about user traffic profiling in general, together with traffic collection and an overview of processing and monitoring tools, as well as a review of network traffic analysis techniques. Chapter Two also describes in detail the challenges, limitations, advantages and tools currently used in user and application traffic classification and behaviour profiling. The last section of the chapter focuses on analysing the state-of-the-

art research on user profiling based on network application traffic monitoring, concluding with the advantages, disadvantages and limitations.

Drawing on the current research, Chapter Three explains the main concept of the proposed method of user identification and behaviour profiling using application-level session timing resolution. The chapter includes an overview of the processing components of the novel proposed method, as well as an explanation of the proposed timing features for identifying users based on network application activity and describing user behaviour.

Chapter Four presents the methodology and data collection for the novel proposed method described in Chapter Three. The chapter explains the pre-processing pipeline and experimentation procedure for the datasets used to validate the proposed methodology, including the timing features identified.

Chapter Five validates the performance of the novel user identification and behaviour profiling using application-level session timing resolution features extracted by utilising the traffic data collected from 23 real users during a 60-day period. It also explains the process used to evaluate the performance and uniqueness of the features extracted and the classification process. The evaluation also includes an analysis of how identifiable and unique user behaviour profiling is over time in the context of the dynamic, changing nature of network applications. This includes an investigation of the effect of different time resolutions (hours and quarter hours) when processing the collected data.

Chapter Six outlines the overall architecture of the user behaviour profiling system using application-level flow sessions timing resolution that can be used to interconnect the proposed method and its components, and provides a form of interconnection that can be used by an investigator in the real world.

Chapter Seven presents the main conclusions of the research, highlighting its main achievements and limitations. Future research opportunities and directions of this project are also discussed.

# **Chapter Two**

## **User and application traffic classification**

## **2 User and application traffic classification**

### **2.1 Introduction**

In today's technology-driven world, user profiles are a translation of each user's activity and include a range of user information, such as data on personal interests and preferences (Cufoglu, 2014). Identifying a user from a flow of network traffic information and then profiling that user's behaviour is a challenging task for researchers due to the complexity and growth of internet traffic and dynamic changes to internet protocols. User identification, which leads to user profiling, helps the ISP by providing information about what an individual is doing, as user behaviour changes and adapts and the online interaction environment also changes (e.g., in terms of website browsing, which applications are browsed and for how long) (Xue and Dong, 2013). In the attempt to understand users' behaviour from application traffic, classification leads to the identification of usage trends, which assists in profiling user traffic. However, profiling individual user traffic in real-time network resource management remains challenging due to variations in user traffic behaviour, leading to repeated updates in the network topology (Vinupaul et al. 2017; Heer & Chi n.d.2006). In profiling an individual user from network traffic, user information is extracted from applications, leading to the identification of meaningful statistical traffic patterns that assist in creating user behaviour profiles for several purposes, ranging from network security to online trend analysis (Chi et al. 2002; Paredes-Oliva et al. 2012).

A network profile is an inventory of information on a network and its associated purpose that shows a comparison between users' network traffic profiles in order to enhance the organisation's security and policy perspective. A behaviour profile can also enable a network administrator to decide and configure the changes that will affect a user on the network. Furthermore, business administrators can use such profiles to help establish a

long-term plan to manage the organisation's network policy (Yang, 2010). This chapter presents background information on user and application traffic classification. It begins with an overview of the sources of information, network monitoring tools and analysis techniques for network traffic classification. The next part describes in detail the current state-of-the-art research in the areas of network traffic classification and user identification and behavioural profiling, including challenges and advantages. The last section of the chapter discusses and summarises the achievements and limitations of the existing research on user profiling based on network application traffic monitoring.

## **2.2 User profiling**

This section provides an introductory context for defining the network traffic analysis concept and the terms that are utilised when dealing with network traffic metadata. Traffic analysis is used to assist investigations of specific network traffic problems, such as traffic characterisation and classification, application detection and user profiling. Traffic analysis is utilised to examine raw network traffic and process it in order to produce statistics and look deeper into what type of network traffic packets/flows are flowing through the network. This is done for the purpose of performance, security, policy organisation, user profiling and general network management (Lucas, 2010).

Traffic characterisation is mainly used to examine network traffic to understand the traffic flowing through networks. A network contains a huge volume of packets, both bulk traffic and regular traffic, which can be used to investigate what is happening in the network and to characterise the structure of the packets, which is helpful in managing the volume of traffic and dealing with network devices (Veres and Ionescu, 2009).

Traffic classification is mainly utilised for network resource management in analysing real-time network traffic metadata, such as video streaming, and non-real-time traffic, such as email. The classification of network traffic can be used in many applications, such

as application and user identification and quality of service and intrusion detection systems (Piskac and Novotny, 2011). Therefore, the importance of traffic classification lies in the priority of some traffic versus other traffic and in managing the resources on a network by classifying and monitoring the traffic transmitted.

Application detection is a way of identifying the commonly used applications on an investigated network. Application detection attempts to identify the type of application by going a step further with traffic classification to detect not only the real-time traffic, such as video streaming, but also to look specifically at news and social media (Alcock and Nelson, 2012). Therefore, application detection investigation is a technique utilised to identify an application from different perspectives, such as intrusion detection and network security, as network security has become an important field due to the increasing number of devices used and the huge amount of traffic investigated.

User profiling covers all areas by trying to investigate an entire network by identifying users based on their usage, as well as investigating how individual users interact and browse. User profiling is utilised to determine a user's activities and behaviours and could be used in different applications, such as security, anomaly detection, and monitoring and managing network traffic (Cufoglu, 2014). As a result, more research and investigation are needed to identify and profile users by monitoring and analysing network traffic based on an application-level footprint in order to extract novel features representing users' activity. Examining users' profiles can help the network administrator and ISP to make informed decisions in relation to security, organisation policy, and managing the network traffic.

### **2.3 Traffic collection and processing**

This section provides an introductory context to the current traffic collection and processing approaches; in other words, the tools used to capture and store network traffic

information based on a packet capture (pcap) and flow (NetFlow) format. As such, the next section (2.3.1) provides an overview of network monitoring and analysis tools, as well as the open-source NetFlow capturing and analysis tools used in this project. Traffic travels through the network and can be captured and analysed and there are a number of options for capturing or summarising it. Traffic is made up of packets and can be meaningfully grouped into flows and connections, which are implemented in the following formats: IP source, IP destination, port source, port destination and protocol information. All packets and traffic can be captured and stored, and pcap is one of the formats that do that. NetFlow is one of the options that can be used to capture and store traffic without saving the content of the packets, although it only does the packet accounting per flow (Li *et al.*, 2013).

NetFlow records information and summarises the network traffic flow as the source and includes the destination IP addresses, source and destination ports, transport protocols (Transmission Control Protocol [TCP]/User Datagram Protocol [UDP]), as well as the traffic information transmitted during a flow session. NetFlow provides an efficient key set of services for IP applications, including network traffic accounting, usage-based network billing, network planning, security, denial-of-service monitoring capabilities and network monitoring. NetFlow also provides valuable information about network users and applications, peak usage times, and traffic routing and contains a flow cache element, which tracks the number of bytes and packets belonging to similar traffic during a certain period until the cache expires. Then, the NetFlow collector receives this information from its exporter and stores it as NetFlow records at a single IP source and destination. In addition, the flow is the set of IP packets passing an observation point in the network during a certain time interval; all the packets in the flow have a set of common properties that are stored for later analysis (Wang *et al.*, 2011).

Raw traffic information (pcap) is another source of information and can be used for counting NetFlow in terms of packets per flow. Pcap consists of an application-programming interface for packet capturing on the network. The pcap implemented on Linux and UNIX is the libpcap library, which is a standard packet capture library. Libpcap was developed to work with the Berkeley Packet Filter (BPF) kernel device. BPF is an operating system (OS) kernel extension that enables communication between the OS and the network interface card (NIC). Libpcap is a C language library that extends the BPF library constructs and is used to capture the packets on the network from the network adapter (Qadeer *et al*, 2010).

A flow-based approach has been used in user profiling based on network application traffic by using the sampled NetFlow traffic information (IP source/destination, time information, transport layer information) as the profiling parameters to be analysed. Bianco *et al.* (2009) published a paper regarding the steps involved in creating a profile in a mid-size to large network that serves hundreds to thousands of users. Bianco *et al.* (2009) attempted to build a profile for the analysis of traffic information (e.g., ports, protocols and other sampled NetFlow records). The framework used flow-based traffic profiling and performed best with a small number of users. It consisted of different stages, starting from initiating a user behaviour pattern, including the number of sessions, overall user behavioural patterns, retrieving patterns for all the users, and measuring the distance between two profiles. The classification method achieved 90% accuracy. Regardless of the promising accuracy rate, however, the study has several serious issues that may affect its reliability, particularly as the authors evaluated the approach with just eight users from a large dataset containing 70,000 users. Bakhshi and Ghita (2016b) proposed a network monitoring technique to identify and predict user behaviour using NetFlow and cluster analysis. Two hundred users were clustered on six unique traffic profiles. The six user traffic profiles clustered by Bakhshi and Ghita (2016b) were High- and Low-intensity

surfers, All-rounders, Communicators, Concealers and Downloaders. Each profile was created based on a mix of activities: Browsing, Emails, Downloading, Streaming, Games, Communications, and Unknown. The results were predicted from a four-week observation and analysis of the traffic and indicated that the profiles were static with changing the probability of the device to change profiles between 3-19%. Although the static rate is promising, the study was aimed at clustering users, rather than establishing individual user traffic profiles.

The two research papers above have demonstrated the usefulness of NetFlow in flow-based user profiling that uses sampled and derived features extracted from the NetFlow records. Furthermore, NetFlow records were used in this project by extracting new features from the basic NetFlow records. Moreover, the timestamp attribute was used in identifying users based on application-level traffic information to improve the diversity between users in different applications. NetFlow traffic information is important in terms of reducing complexity but is time consuming with regard to the comprehensive view it provides of user sessions and application usage.

### **2.3.1 Network monitoring tools**

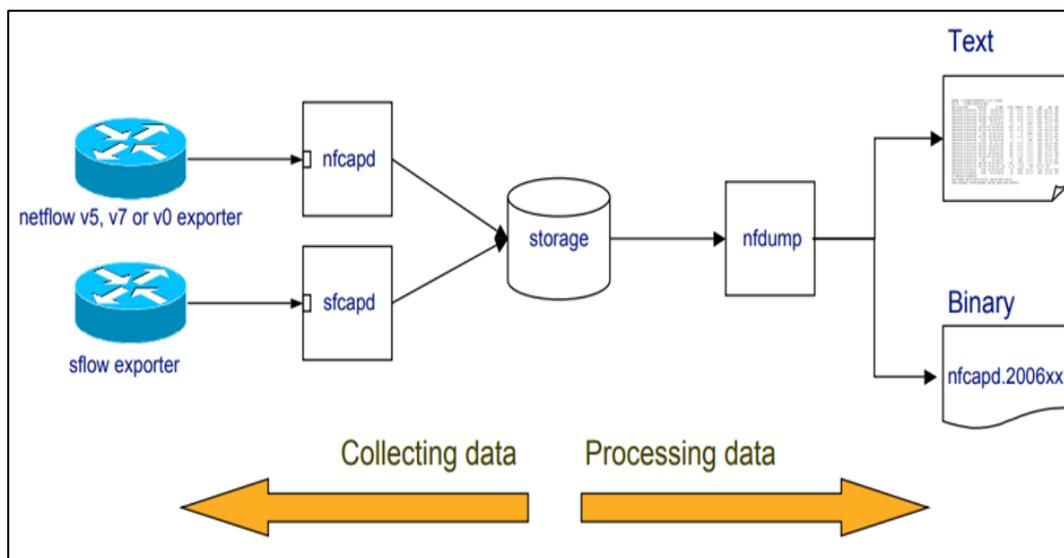
This section introduces the context of the current network monitoring and analysis tools and includes an overview of some of the tools' components and the advantages of these components in being utilised with the tools examined. First, tcpdump is a common tool that uses the command line to capture packets and to filter and analyse network traffic. Tcpdump allows the user to display the TCP/IP and another packet being transmitted or received over a network channel to which the computer is attached and display deep payload information. Tcpdump works on UNIX, Linux, and Solaris, BSD, OSX, HP-UX, and Android. In these systems, tcpdump uses the libpcap and WinPcap library to capture packets; the tcpdump port for Windows is called WinDump. Tcpdump reads packets from

an NIC or from previously created and saved packet files. Tcpdump can also write packets to a standard output or file (Medhi n.d.2004).

A second example is NetFlow, which is used to access and analyse network traffic and gives the network administrator the ability to enhance the security of the network in addition to network monitoring. The network administrator can use NetFlow traffic information to know who, what, when, and where network traffic information is flowing. By analysing the network traffic displayed by NetFlow, a network administrator can obtain traffic information, such as the source and destination IP address of the traffic, the source and destination ports, timing, transport layer, type of service (ToS) and interface information. Figure 2.1 shows the open-source nfdump/softflowd process that was used in this project.

NetFlow consists of three main components:

- **Flow exporter:** this aggregates packets into flows and exports flow records towards one or more flow collectors. An open-source example of a flow exporter is softflowd.
- **Flow collector:** this is used to collect, store and pre-process flow data received from a flow exporter; the flow data saved as NetFlow records are an open-source example of the flow collector nfcapd.
- **Flow analysis:** this component is used to analyse the NetFlow records traffic information received in the context of network traffic analysis, such as traffic monitoring and traffic profiling; an open-source example of a flow analyser is nfdump (Cisco, 2016).



**Figure 2.1: NetFlow (nfdump) architecture (Haag, 2006)**

Third, Wireshark is the most commonly used network analyser and is available free as an open-source tool. Whether connected to a wired or wireless network, Wireshark processes traffic using a libpcap or winpcap link-layer interface. The wiretap library contained on Wireshark can read a variety of trace file formats, including tcpdump libpcap (Chappell 2012).

Wireshark is similar to tcpdump but has a graphical front-end, as well as integrated sorting and filtering options. Wireshark allows users to set the network interface controllers that support a promiscuous mode so that they can see all traffic clearly on that interface, not only traffic addressed to one of the interface's arranged addresses and broadcast/multicast traffic. However, when capturing packets in promiscuous mode, not all traffic through the network is sent to the interface where the capture was done; therefore, capturing in promiscuous mode is not necessarily sufficient to see all network traffic. Tcpdump is more flexible than Wireshark because it uses the command line to collect and analyse network traffic (Qadeer *et al.*, 2010).

## 2.4 Analysis techniques

The next two sections (2.4.1 and 2.4.2) set the context for the network traffic analysis techniques and provide an overview of some of the algorithms and tools utilised in these techniques. The first section provides an overview of deep packet inspection (DPI) techniques and some of the analysis tools used in the field of network traffic analysis. The second section provides an overview of the machine learning (ML) classification technique and its two main types: supervised and unsupervised.

### 2.4.1 Deep packet inspection

DPI is the state-of-the-art technology for traffic classification. Although DPI is one of the most-used classification techniques, its actual performance is still unclear among the research community. The most commonly used DPI tools in traffic classification are open DPI, Layer 7 (L7)-filtering, nDPI and Libprotoident.

- **OpenDPI:** this is an open-source classifier that was derived from an earlier version of PACE (Bujlow, CarelaEspaol and BarletRos, 2015) by removing support for encrypted protocols, as well as all performance optimisations.
- **L7-filter:** this was created in 2003 as a classifier for the Linux net filter and can recognise traffic on the application layer (L7). The L7-filter was developed as a set of rules and a classification engine that can be used independently of each other.
- **nDPI:** this is a fork of OpenDPI which can be optimised and extended with new protocols. It supports many encrypted protocols due to its ability to analyse session certificates. nDPI supports more than 100 protocols.
- **Libprotoident:** the C library introduced lightweight packet inspection, which examines the first four bytes of a payload in each direction (Bujlow, Carela-Español and Barlet-Ros, 2015).

A network packet contains various data fields, such as the header, trailer and payload. The packet-based network analysis method known as DPI inspects payload data to discover the data packets transmitted through the network. Many research papers have used DPI to identify the applications used, thereby providing the information needed to build a profile (Liu and Wu, 2013; Finsterbusch *et al.*, 2014).

Packet-based network analysis has been used for various network-related traffic analyses, such as traffic profiling, classification, measurement and management. Of the different methods being devised to improve performance, the packet-based approach is still computationally expensive, due to the nature of inspecting the payload of the packet, as an analysis of each packet transmitted through the network is required (Auld, Moore and Gull, 2007; Megyesi and Molnr, 2012).

This section has explained that the DPI tool L7-filter has the ability to classify and differentiate between application traffic information, which could be utilised in the next stage of the research to identify and profile user behaviour correctly based on the application traffic (Dainotti, Pescapè and Claffy, 2012).

#### **2.4.2 Machine learning analysis**

Machine learning is a process that allows computers to learn directly from examples and experience in the form of data to solve a specific problem. The goal is to devise learning algorithms that conduct learning automatically, without human assistance. ML has been used in web search, spam filter, recommender system, credit scoring and user traffic classification (Wagner *et al.*, 2011).

ML technology has long been recognised as an alternative to traffic classification (Alcock and Nelson, 2012) and its techniques are divided into supervised and unsupervised (Nguyen and Armitage, 2008). A gradient boosting algorithm was used in this project to classify and profile users based on application usage (*Bivens et al*, 2002). The following

paragraphs provide an overview of the two types of ML technique: first, an overview of the supervised ML that was utilised in this project; and second, an overview of unsupervised ML to illustrate the difference between the two approaches.

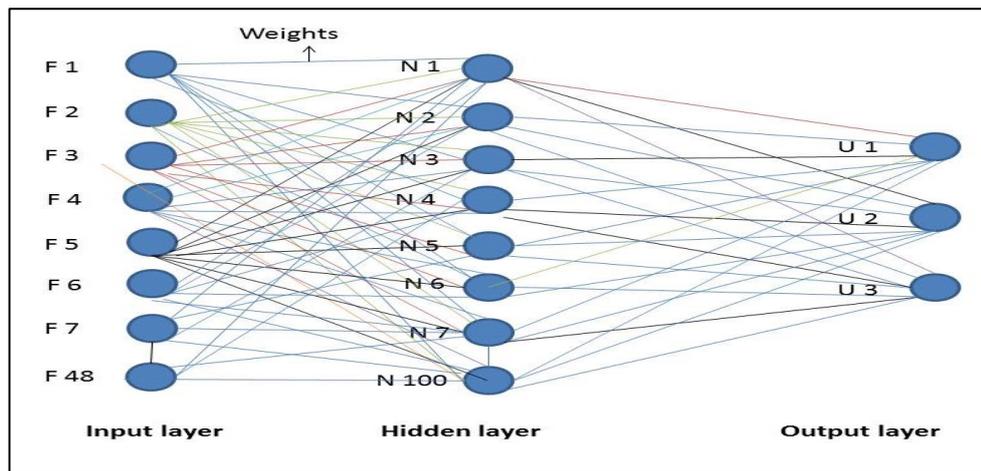
As referred to earlier, this project used a supervised ML technique, which involved predicting a group of users for the data exemplar by adding data as an input and applying an algorithm. Gradient boosting was implemented in the experiments conducted for this project. Supervised ML techniques are known as classification techniques. The idea behind supervised ML is to predefine the classes in order to classify new elements by building a relationship between the input and the output of the classifier. The result predicted by the classifier is called the classification module and the input data are called the dataset. There are two important phases in the supervised ML technique:

- **Training:** the training phase examines the data provided and builds a classification model.
- **Testing:** when the training phase builds a new model, the model is used to classify unseen elements.

An important research direction in supervised ML is neural networks. One advantage of a neural network is that there are fewer requirements for formal statistical training and the ability indirectly to detect complex non-linear relationships between dependent and independent variables (Hong *et al*, 2015). Feedforward multi-layered perceptron neural networks are known for their large-scale problem optimisation techniques, due to their ability to deal with complex problems.

Feedforward neural networks (FFNN) are organised into layers, which are made up of a number of connected nodes or neurons. The input layer of the neural network connects patterns with hidden layers. All the components of the neural networks are connected via core components called weights, as illustrated in Figure 2.2. A basic FFNN contains three

layers, each containing different neurons or nodes. The input layer contains the nodes with the number of features included in the dataset, which means that if the dataset contains 200 features, the input layer will contain the 200 nodes to be transmitted between layers. The second component of the FFNN is the hidden layer and is the most important. The hidden layer provides necessary discrimination for separating the training data, as increasing the number of neurons in the hidden layer decreases the errors in the training data, as well as reducing the amount of generalisation, which is important, as the effect of the performance of the modules depends on the problem. The hidden layer ensures and only passes on important information from the input layer to the output layer. Neural network layers are based on connections that have a weight value. These weight values change depending on the training phase in the problem under investigation to reduce the errors between the layers connected in the neural network. The last layer of the FFNN is the output layer, which takes the input from the hidden layer after the calculation of the weights of the connections between the input layer and the hidden layer. Given that, the output layer can obtain a limited number of neurons depending on the training of the module and the percentage of errors that occur between the input layer and the hidden layer.



**Figure 2.2: Neural network structure and weights**

If a neural network is properly designed and implemented, it can classify the various behaviours of users from the network traffic. One of the main advantages of neural networks is their ability to infer solutions without knowing the regularities in the data (Kumar and Sachdeva, 2010). For instance, Bivens *et al.* (2002) detected intrusions based on users' generic network traffic by using supervised learning of neural networks at different time intervals.

Another important research direction utilised in several studies (Zhang, 2008; Phan, Sun and Tay, 2017), including this project, is gradient boosting supervised ML. Gradient boosting is categorised as one of the decision tree families according to several parameters, such as being tree-specific, and boosting is used to convert weak models into strong module by weighting the small tree outcome based on the previous tree. The strongest point concerning gradient boosting classifiers is that they give a correctly predicted class a low weight and a misclassified one is given a high weight (Dainotti, Pescape and Claffy, 2012). Gradient boosting is a useful practical supervised ML approach for different predictive tasks and can dependably provide more accurate results than straight single ML models. This approach is inspired by the gradient boosting framework previously applied to solve classification and regression problems and more recently to train conditional random fields. The boosting of supervised ML is done to build a series of

small decision trees based on the data collected, each tree attempting to correct the errors from the previous stage. During the last few years, many practical studies have been published that use decision trees as the basis for learning about gradient boosting (Garsva *et al.*, 2014). Furthermore, the algorithm can optimise any differentiable loss function by using a gradient descent approach. This approach builds the trees sequentially to sum up an individual tree that provides the best solution under different conditions.

The other type of ML, unsupervised learning, is known for its clustering algorithms. The clustering algorithms do not classify elements by using predefined classes, as implemented in supervised ML techniques. Instead, the clustering technique groups the elements into clusters (groups) by finding the similarities among the elements of the input data by measuring the distance between them, using a metric such as Euclidean distance (Dainotti, Pescape and Claffy, 2012).

K-means is a well-known unsupervised learning algorithm. K-means randomly generates k clusters, which represent the clustered elements. Depending on the similarity between the elements, all new objects are assigned to the cluster of elements that are most similar to them. The process continues until all the objects are assigned to their nearest neighbour cluster (Singh, 2015).

Unsupervised learning does not require labelling, as implemented in supervised ML. However, semi-supervised learning utilises both labelled and unlabelled data for training the dataset. Clustering is a form of statistical data analysis that organises a dataset into individual clusters: element groupings the membership of which is determined by a shared similarity.

In general, clustering algorithms appear in one of two forms: agglomerative or partitioning. Agglomerative algorithms work hierarchically, first merging elements of highest similarity, and then recursively merging clusters until the desired level is reached

(Singh, 2015). The resulting multidimensional data are handled in a dynamic environment and ML is used in various sectors, such as financial departments, banking and healthcare systems. Amazon and Google use ML to solve classification and regression problems, as well as directing advertisements based on user behaviour (Bermudez and Mellia, 2012; Afridi *et al.*, 2018). Therefore, neural networks and gradient boosting supervised ML were deemed suitable to solving the classification problem investigated in this project.

## **2.5 Previous work on user profiling based on network application traffic**

This section considers the literature on user profiling based on network application traffic monitoring, such as the traffic classification techniques that are utilised in port-based, payload-based and statistical ML techniques. The following section (2.6) provides an overview of user identification and anomaly detection techniques. The last two sections of the chapter discuss and conclude the context of the literature reviewed in this section.

### **2.5.1 Network traffic classification techniques**

A lot of work has been done on network user and application classification in order to classify network traffic into categories that can then be utilised to solve a number of classification regression problems (Williams, Zander and Armitage, 2006). Network application traffic classification has limitations in payload- and port-based techniques because of the randomly assigned port numbers of certain applications and encryption, which affects the decisions in an investigated problem. Approaches that are statistically based provide alternative methods for solving classification problems, as they eliminate the limitations that occur with the previous classification techniques (Bonald, 2015). Behaviour profiling has been used in a range of areas, such as user identification (Banse, Herrmann and Federrath, 2012) and anomaly detection, to investigate a user's pattern in the network, which could help with issues such as security, monitoring and organisational policy. Therefore, many identification and anomaly detection problems need user and

application traffic classification to be investigated further. As anomaly detection is carried out based on the desired or positive behaviour of users and processes, user behaviour profiling has been investigated to distinguish between normal behaviour and observed anomalies (Lim and Jones, 2008). Identification of a user based on network traffic flow patterns can be challenging, however, due to the presence of noise in the network traffic data investigated. Melnikov (2010) discusses the potential for inferring user identity from network traffic metadata distribution and the cross-correlation of various trace parameters and the relationships between the flows. Although the results reported were preliminary, with a 60% level of accuracy, additional research may yield results that are more promising if the method can be enhanced. However, extracting a set of features to enable users to be identified from the application level is a critical step in investigating how a user's activity changes over time, as this affects the stability of the result (Kumar, 2014).

### **2.5.2 Port-based techniques**

Port-based techniques are the oldest and most common method of traffic classification. Port-based techniques depend on the analysis of the communication ports contained in the TCP/UDP port numbers, the ports for the protocols being assigned by the Internet Assigned Numbers Authority (IANA). Nevertheless, not all protocols can be classified using port-based approaches. For example, the peer-to-peer system and the File Transfer Protocol (FTP) can use random ports and, for this reason, a port-based approach is not sufficient for traffic classification.

Erman et al. (2006) proposed a method of user traffic identification using unsupervised ML and compared it with supervised ML (a neural network). They analysed a dataset of publicly available traces for connection identification using the statistical flow information between two nodes, which was identified based on the IP address and the transport-layer port numbers. Once a connection had been identified, they performed a

calculation of the total number of packets, mean packet size (mps), mean data packet size, flow duration, and mean inter-arrival time of packets. Erman et al.'s (2006) results show that unsupervised clustering achieved an accuracy of more than 90% of users identified based on the clusters assigned, which was higher than the neural network's accuracy of 85%. In addition, the clustering reduced the time consumed by classifying the connections based on the publicly available dataset used. However, using a well-known port number is not sufficient because many well-known applications have specific port numbers and many use dynamic port-negotiation mechanisms to hide from network security tools, which prevents discovering and identifying the traffic from the applications used by the user.

Kounavis et al. n.d (2003) proposed a method to exploit the structure and characteristics of packet classification using port numbers to identify user behaviour based on an application-level classification. The analysis method yielded three key findings (rules). The first rule was that the investigated datasets contained the source/destination IP addresses that represent the user's network traffic metadata and the transport-level fields (e.g., source port, destination port, s2d packets and destination to source d2s packets, s2d bytes and d2s bytes), which represent the network application information. The second rule is that IP address pairs identify two connections that overlap each other. The third rule is that only a small number of transport-level fields are sufficient to characterise the datasets of different sizes. For these three rules, the researchers used the source IP address, destination IP address, source port, destination port, action and priority, but this is not sufficient, as popular applications use well-known port numbers, such as HTTP port (80) and FTP port (21); for this reason, statistical and ML techniques should be used to address the problem of random port numbers.

### 2.5.3 Payload-based techniques

Payload-based techniques, or DPI, are an important set of techniques used in traffic classification (Finsterbusch *et al.*, 2014). This is a very useful approach for going deep into the content of a packet to know the details of what the packet carries, as well as the timestamps with which the packet starts and ends. On the other hand, many research papers have used DPI in combination with port- and statistical-based techniques to attain more accuracy and robustness (Zhang *et al.*, 2013). However, the problem with not using this technique in combination with other approaches is that encryption prevents the analyser from going deep into the content of the packet. Furthermore, DPI has been used in combination with statistical techniques to identify user activity from the user traffic information and application usage (Deri *et al.*, 2014). Many research papers have proposed DPI tools that have been used to carry out payload-based techniques automatically, such as nDPI, L7-filtering, and BLINC (Karagiannis *et al.* 2005; Bujlow *et al.* 2015).

DPI has been used to investigate user behaviour based on classifying users on web applications to allow the ISP to recognise and identify users through the network. The headers of the TCP/IP packets, which contain addresses, port numbers and information content created by the user, are then inspected (Bujlow, Carela-Español and Barlet-Ros, 2015).

DPI platforms have evolved from static rules-based filtering engines to sophisticated enforcement points allowing packet and protocol classification, prioritisation and shaping (Wang, An and Yang, 2011). DPI has been applied in enterprises and telecommunications networks and utilised in various areas, such as traffic management policy enforcement, adult content filtering, lawful interception and behaviour profiling (Xu, Zhang and Bhattacharyya, 2008).

DPI relies on a strong classification engine to monitor data traffic and classify each packet based on port, protocol, interface, origin and destination (Szabó *et al*, 2011). Furthermore, DPI is a more sophisticated engine that goes beyond layer 3 and can distinguish classes of traffic using headers. The DPI classification engine is also sufficient for most types of traffic inspection, such as web browsing, email, Voice over Internet Protocol (VoIP), video conferencing and peer-to-peer sharing (Nascimento *et al*, 2014).

#### **2.5.4 Statistical-based techniques**

Statistical techniques are the most common method of network traffic classification. Statistical-based techniques are generally applied to flow-based and packet-based approaches as a pre-process before applying ML to network traffic metadata for evaluation. NetFlow data contain statistics regarding the network flow generated and exported from the routers. Statistical approaches are usually easy to implement and provide accurate results. However, these approaches are only suitable for known cases and lack the ability to adapt to new cases.

Balram and Wilscy (2014) proposed a bot detection mechanism for a single host and user traffic profile for filtering out the normal traffic generated by the host. The method was used to examine user IP addresses and the IP destination of the transmitting network packet used in Google (web searching) to identify user behaviour activity. A profile of normal behaviour regarding user traffic was generated by utilising the IP destination and domain name on outgoing flows. The dataset in Balram and Wilscy (2014) was built by collecting the network traffic generated by 50 users, which contained the regular 5-tuple of NetFlow records. The flows that contacted the destinations in the normal profile were considered ‘innocent’. As a result of the mechanism used in this paper, Balram and Wilscy (2014) found the approach to be valid for detecting bot traffic and the destination contacted by the bot. Furthermore, the results show a detection rate of 92%, with false

negative (FN) rates of 8%. Although the mechanism used in the study is relevant to the current research, adding more applications and a deeper investigation into the IP destination, in addition to the Domain Name System (DNS), was deemed by the current researcher to be more accurate and able to extract more features to make a statistical dataset that would give sufficient results for behaviour profiling patterns.

Kamesh and Sakthi Priya (2014) aimed to investigate the classification of behavioural performance that is affected by traffic morphing. Traffic morphing is a collection of traffic protecting tactics, the aim of which is to frustrate behaviour-based classification. It usually involves applying a set of faking strategies, such as padding packets, and altering the inter-arrival time on network traces to protect traffic from being recognised by behavioural features. The features used in this study were created by taking the first packet of the flows and ignoring the small data packets (such as SYN, the synchronise message, and ACK, the acknowledgement message) because these packets were deemed irrelevant to the application layer. Moreover, Kamesh and Sakthi Priya (2014) utilised a method of extracting flows that contained an inter-arrival time threshold of less than 20 to 200 ms to provide enough flow instance datasets for training the classifier, as well as to represent non-trivial network traffic components. Then, the flows with a low number of predefined packet thresholds were ignored - set as 10 in this work - the idea behind which was to make the dataset resemble real-world network environments. While traffic morphing gives the user more privacy, meaning that he/she can be protected from a classification attack, the inter-arrival time threshold utilised in this study is sufficient to analyse the traffic based on the timing features and assign a threshold to combine the flows as sessions or windows. However, assigning flows that contain just 10 packets might affect the evaluation of the system being investigated. Therefore, the flow inter-arrival time was used in this project to extract the timing features to identify the user based on an application-level timing resolution.

In other research, Tao et al. n.d. (2015) aimed to examine typical user behaviour activity and present a traffic identification method for describing the features of that behaviour. User network activity was divided into different sequences with various frame sizes and inter-arrival times in order to differentiate users' activities. A direct allocation model was used to extract potential means of characterisation from the activity sequences. In addition, to investigate the relationship between user activity and traffic information, parameters were extracted from the packets' inter-arrival time and frame size. User activity is subject to end at some point, and the duration of this activity can be used to predict a number of interactive sub procedures (SPs). Each SP predicted from the user session duration time has shorter units, called traffic elements (TEs). Moreover, the packets' inter-arrival times and frame sizes for each trace file were converted to vector sequences. After that, to differentiate between user activities, the vector sequence and SPs were calculated before applying the clustering algorithm to the dataset. Tao et al. n.d. (2015) used three ML algorithms: Naïve Bayes, multi-layer perceptron and K-means clustering. The results of the identification of mixed activities show that the multi-layer perceptron scored more than 60%, the Naïve Bayes about 60% and K-means clustering about 50%. The research showed the possibility of identifying user online activity through frame size and inter-arrival time and without packet-payload inspection. However, identifying user behaviour based on network application usage needs to have a relation between the time stamp and the connections that are available on the packet payload. The study scores low accuracy due to the quality of the features used on the datasets and the number in the sample. The study focused on inter-arrival time and frame size, which might be close to the idea of the sessions and flow-inter-arrival time method utilised on this project. However, there is a difference, in that Tao et al. n.d. (2015) used the packet level; in the current project, the flow level was used.

### 2.5.5 Machine learning

This section presents the state-of-the-art research regarding ML algorithms that have been used in the field of statistical-based techniques, as well as the type of features extracted for traffic analysis using ML algorithms. A lot of work has been done on traffic analysis using ML, which represents a collection of methods for discovering knowledge by searching for patterns. The core learning types in machine learning are listed below.

- **Classification:** a method for classifying inputs to labelled outputs.
- **Clustering:** a method for grouping inputs into clusters.
- **Association:** a method for discovering interesting relations between the features in datasets.
- **Prediction:** a method for predicting the outcome regarding a numerical value.

ML schemes include information theory, neural networks, support vector machine (SVM) learning, genetic algorithms, and many more. ML algorithms need a collection of features for extraction and selection is critical. The features extracted must be tested for their appropriateness, depending on the problem and the classification methods applied.

The following studies have applied a set of traffic classification techniques that are flow- or packet-based by extracting a set of statistical features and applying ML. Many studies have used supervised ML algorithms (*Jiang et al., 2007; Carela-Espanol, Barlet-Ros and Solé-Pareta, 2009; Rossi and Valenti, 2010*) to classify network traffic for a range of purposes, such as user identification and application detection, as well as using features extracted from the flow level (NetFlow records). On the other hand, many other studies have used unsupervised ML (*Qin et al., 2014; Theriault et al., 2015*) to classify network traffic for various purposes. However, analysis of the following studies showed that the performance of supervised ML approaches was higher than that of unsupervised approaches, especially when the number of investigated users was low. The number of

users plays an important role in the network investigated, depending on the purpose of the investigation and the relevance of the extracted features, as some approaches produced acceptable performance although only up to 20 users were mentioned (Carela-Espanol, Barlet-Ros and Solé-Pareta, 2009; Rossi and Valenti, 2010). However, other approaches had a low performance compared with others that used a similar number of users (Auld, Moore and Gull, 2007; Zhang *et al.*, 2011).

Most of the studies considered utilised basic statistical features, such as number of packets, size of packets, and number of connections (Zhang *et al.*, 2011; Malott and Chellappan, 2014), and features were used for different purposes; mostly to classify applications, which is not what the current study is looking to achieve. Indeed, this research focused on the extraction of statistical features from the flow level in order to identify users based on network application traffic, which is a more challenging classification problem. The inter-arrival time has been utilised by several studies (Jin *et al.*, 2012; Suznjevic, Skorin-kapov and Humar, 2014) for different purposes, as well as in this research, and the flow inter-arrival time for the analysis of the flow sessions.

Previous studies (Banse, Herrmann and Federrath, 2012; McDowell, 2013) have reported that using DNS queries is more accurate than using IP destinations for filtering connections in network traffic analysis. Therefore, filtering connections based on DNS queries has its strengths and limitations, depending on the services investigated, as some belong to the same servers; for example, YouTube is related to Google servers, which complicates the differentiation of services (Kumar, 2014; Potdar, S. and D., 2017). The following section analyses the approaches utilised and their limitations and presents trending research papers based on extracting relevant features using ML algorithms to classify traffic for different purposes. The relation between the work conducted for this project and previous research is also considered.

Jiang et al. n.d. (2007) focused on one of the essential network measurements: primitive network traffic classification based on NetFlow records. In the study, the NetFlow records were utilised in building an effective network management system. Network traffic data were collected from a large network that had 1,000 users over a period of 24 hours for network traffic classification and network management and a Naïve Bayes supervised ML algorithm was applied to classify generic network traffic. The Naïve Bayes classifier is one of the most commonly used ML algorithms and calculates by counting probabilities of frequencies in the dataset investigated. Naïve Bayes is also one of the fastest algorithms for training a dataset, which makes the performance of the module slightly higher, with less bias and overfitting of any unseen test data, and improved the performance of the dataset investigated in this paper. In addition, the features were extracted from the NetFlow tuples and derivative features were added to the NetFlow tuples, as shown in Table 2.1. The accuracy of the results reached 91% and the study demonstrates that NetFlow records (basic and derivative features) can be usefully employed in classifying user and network traffic information. The interesting aspect of this work is that the researchers used IP information and derived features, such as the duration of the user session, packet rate and flow rate, which gives a comprehensive view of a user’s activity; however, more investigation into the DNS is necessary to classify network traffic for managing a network.

**Table 2.1: Basic features and derivative features in flow records (Jiang et al., 2007)**

<b>Basic features (NetFlow)</b>	<b>Explanation</b>
Srcip/dstip	Source/destination IP addresses
Srcport/dstport	Source/destination port numbers
ToS	IP type of service
S time/e time	Flow start/end timestamp
TCP flags	Cumulative OR of TCP flags
Bytes	Total number of bytes observed
pkt	Number of packets observed
<b>Derivative features</b>	<b>Explanation</b>
Length	Duration of flow (end time – start time)
Packet size	Average packet size (bytes/packets)
Byte rate	Average flow rate (bytes/length)
Packet rate	Average packet rate (packets/length)
TCP flags	Flags = syn/ack/fin/rst/psh/urg flag

Carela-Espanol et al. (2009) focused on applying the basic sampled NetFlow tuples to one of the most common ML methods to analyse the impact of traffic sampling on NetFlow performance and address the traffic classification problem. They attempted to apply flow-based instead of packet-level traces, as the main consistent feature when using NetFlow is the limited amount of information available for use as features in ML-based classification methods. The dataset consisted of 40 users and the system was evaluated with 15 features extracted from the sampled NetFlow data. The network traffic data were collected from a large university network and the dataset contained two phases: flows and the labelled applications, where flow is represented as a vector of features and an application is a label that identifies the network applications generated by the flow. The authors used the C4.5 ML algorithm in their paper and achieved 90% accuracy. The C4.5 is an older decision tree algorithm that is built using a basic top-down greedy decision tree, which increases its ability to solve classification and regression problems. The C4.5 improved the performance of the module in the dataset investigated in this work because of its high level of precision and the ability of decision tree approaches in general to deal with weak modules. However, the work sampled NetFlow records without extracting the relevant user session duration (end time - start time), which is important in terms of identifying a user profile from network application traffic and grouping the user activity with application behaviour. In addition, L7 filtering, which is based on the DPI technique, was used in the study and combined with a statistical-based technique. However, the combination of DPI and statistical-based techniques might have affected the performance during the evaluation of the system. The use of a flow-based technique to extract basic and derived features could have provided helpful criteria for applying to user profiling and identification in this project.

Rossi and Valenti (2010) proposed an algorithm that is different from the two modules presented previously to investigate user profiles and successfully utilised NetFlow

records for user profiling and traffic classification. Their proposed classifier identified an application using a simple count of received packets and bytes, and the results show that the methodology employed correctly identified byte-wise traffic volume with an accuracy of 90%. The features used in this paper extracted a key composed of specific features from each flow (NetFlow v5 is classically 5-tuple). In addition to the classical tuple, the researchers also used a number of attributes, such as cumulative packets, byte counters, flow start and finish timestamps, IP ToS, TCP flags, MPLS labels and physical input/output interface indexes. The count of connections utilised might be useful in the case of investigating and analysing traffic to identify the user.

Qin et al. (2014) proposed a method for investigating users' behavioural similarities that employed a user-clustering algorithm at various network prefix levels. First, the network traffic characteristics were modelled using bi-directional flow information and bipartite charts. The profiles of the users' behaviour were characterised by identifying four relevant traffic parameters. Second, users' behaviour similarities were calculated using weight factors, which were selected by employing intelligent entropy gain. The K-means clustering algorithm was used to cluster users using the behaviour similarities calculated earlier from the users' behaviour profile. The traffic parameters used were packets with a definite transport protocol, source port sets, destination port sets, and destination region sets. Finally, the K-means clustering algorithm designed was applied to determine the characteristics of user behaviour clusters at the same network prefix level. However, although the clustering worked with a group of users, it was not effective for individual users.

Therriault et al. (2015) proposed a methodology for enhancing tools of network-level misuse and intrusion detection using behaviour-based clustering. They utilised a technique for making packet header data into clusters based on the similarity of observed

behaviour, in order to validate the utility of behaviour-based clustering for user behaviour data compression and the efficiency of cluster features in identifying similar user behaviour. For example, source IPs were clustered together based on their pattern of destination port usage. Cluster features were then employed to sort the clusters into a prioritised list for assessment. Clustering was found to greatly reduce the total volume of data reviewed; the number of clusters formed over a given time interval is determined to grow slowly with time, as  $\log(t)$  and simple sorting rules are shown to be sufficient in prioritising a suspicious cluster for investigation. The network traffic data were collected from an internet access point and an enterprise gateway. There was a five-hour period for collecting data from the access point and 72 hours from the enterprise gateway, to examine the accuracy of the proposed method. With regard to the features used to do the clustering, the source IP, destination IP and destination port were joined. This joining was called a joint probability table and it was computed by regressing the packet header data in consideration of the time interval of interest. This calculation was implemented with the test data using SPADE, a Snort plugin. The resulting sample joint probability was used by Theriault et al. (2015) to characterise the dataset for clustering. The method employed in this analysis was to cluster together source IPs (SIPs) that have similar destination port (DP) usage patterns, regardless of the actual destination IPs (DIPs) involved. However, the selection of an appropriate distance measure is of ongoing concern, as the cluster information was still affected by the presence of dominant ports due to the weighted nature of the distance metric. Another issue is that the traffic was collected in one direction, which might have affected the accuracy of the behaviour profile. Even so, the clustering of behaviour profiling utilised by Theriault et al. (2015) had the potential to enhance the methodology proposed in this project.

Malott and Chellappan (2014) investigated individual user profile self-similarities based on characterising internet traffic. Multiple application types were used to collect real

NetFlow traffic information logs. Packet inter-arrival time was calculated by an analysis of the start and end timestamp attributes in the NetFlow records. The extracted features were source/destination IP address, source/destination port, destination port, protocol, octet count and the length of the packet inter-arrival time. Similarity in individual user behaviour was investigated by observing 10 application groupings in emails, chats, HTTP and gaming. Seven applications showed 80% self-similarity and three showed 98% self-similarity, which indicates that calculating the inter-arrival time enhanced the performance of the system. As a result, identifying an individual user's self-similarity by calculating packet inter-arrival time between connections might give a higher behaviour similarity between users, leading to accurate user profiles from network traffic information. Therefore, the use of inter-arrival time could be utilised to identify and profile a user based on timing features by dividing the traffic into sessions or windows.

Auld et al. (2007) proposed a method for classifying applications from traffic information using supervised ML algorithm neural networks. The parameters were extracted from the traffic information applied in the supervised ML algorithm without any source, destination IP address or port data. Moreover, the training and testing dataset was derived from one of several packet headers and packet payload information. These techniques were used to classify more applications, rather than using full packets/payloads for traffic classification. The parameters used by Auld et al. (2007) were as follows: number of flow, packet inter-arrival time, size of TCP/IP header, total number of packets, payload size (bytes), effective bandwidth, mean of packet inter-arrival times, and TCP-specific values derived from the top trace. The accuracy of the results using this method reached 95% of the dataset trained and tested for eight months. Overall, this study highlights the ability to identify user behaviour from application usage using statistical extracted information, instead of using user IP and port numbers. However, this method demonstrates that using a supervised ML algorithm, such as a neural network, gives higher accuracy in user and

application traffic identification and classification when employing statistical parameters instead of IP and port numbers. The extracted features could be utilised to enhance the methodology and for user identification and profiling by applying the flow-inter-arrival time and number of connections to extract flow-timing sessions.

In their work, Zhang et al. (2011) proposed an SVM learning algorithm to identify user activity from network traffic and application profiling. Timestamps, packet sizes and flow directions were used to identify users' behaviour and online activity. The traffic parameters extracted to identify and classify users' activity included the average packet size (bytes), with the average packet inter-arrival time for low/high application bandwidth consumption. Zhang et al. (2011) showed that the same application usage varied greatly through different times and environments. Concurrent user behaviour activities were considered in the paper, which means that when a user browses the internet, multiple windows with different applications being browsed at the same time made it difficult to differentiate between applications. As a result, a hierarchical SVM algorithm was used to solve the problem of concurrent applications and identifying user activity. The SVM concept mainly relies on separating classes into groups by finding the hyperplane between the data. Therefore, the SVM classifier has a limitation when dealing with a very noisy dataset because of its reliance on separating the classes, as some groups are misclassified with other groups, affecting the performance of the module. MAC-layer traffic was used by Zhang et al. (2011) to identify and classify user activity by employing datasets in SVM learning and separating the browsing application to achieve a higher degree of accuracy. The accuracy of the results reached 80% in terms of differentiating user activity and applications. However, monitoring the traffic on the networks will be noisy if multiple applications are browsed at the same time, as this makes it difficult to distinguish between different applications. The features proposed in this study might assist user identification and behaviour profiling, as timestamps were used to calculate the inter-arrival time and

then the average was taken. This project also used inter-arrival time based on the timestamp and the flow level instead of the packet level.

McDowell (2013) designed a method for using DNS and destination IPs for user identification and classification based on models of user behaviour. The destination IP address and DNS were compared and the data filtered to increase accuracy in identification. The number of pages visited by the user was analysed by filtering the packets that contain the TCP SYN, the features used on this type of work traffic, snapshot length (bytes), total bytes, total packets, unique source IPs, total destination IPs, unique destination IPs, and DNS. Naïve Bayes and K-nearest neighbour algorithms were used to classify the dataset. The results show that the DNS query performed better than using destination IP addresses. However, DNS query in addition to more traffic information parameters increased the accuracy of profiling users and identifying users' behaviour through the number of pages visited, as in the current study. Therefore, as the DNS query performed better than destination IP addresses, this approach might be useful in the case of filtering applications for the purpose of user identification and behaviour profiling and it is easier to understand text than numbers.

Jin et al. (2012) had earlier proposed a type of supervised ML based on a flow-level traffic classification system as a solution to the enterprise network traffic classification problem. Their system split the classification task into multiple subtasks and applied supervised ML algorithms in addition to weighted threshold sampling, calibration and intelligent data clustering methods to achieve both accuracy and scalability. This method gives high stability with low-level flow rates of only 3% of TCP and 0.4% of UDP traffic. The traffic parameters used were the lower port, high source port numbers, flow duration, average packet size, average packet rate, destination count, TCP flags, lowest port, highest port, number of packets, bytes, ToS, and the number of two bytes. However, they classified

the endpoint by extracting information freely available on the web and classifying and profiling the end host over lengths of time of 1 to 5 min, which are considered short periods.

Du et al. (2013) proposed a novel technique involving port numbers, DPI and statistical approaches to make use of the advantages of each method. This multi-stage approach was able to classify network applications and the user by TCP or UDP, taking full advantage of combining port, DPI and statistical-based methods. Furthermore, the approach was also shown to reduce the impact on network performance by taking advantage of traditional forms of traffic classification, such as port- and payload-based, compared with traditional statistical-based approaches that make it practical in the real network environment. Du et al. (2013) extracted 50 traffic parameters. The features were the number of packets transmitted and received in a session; bytes transmitted and received in a session; maximum and average packet length transmitted and received in a session; the ratio of the number of packets transmitted and received in a session; and the ratio of the bytes transmitted and received in a session.

Bujlow et al. (2012) designed a method of network traffic classification using a supervised ML algorithm: C5.0. The C5.0 is an improved version of the C4.5 classifier, which relies on a basic top-down decision tree classifier. The C5.0 has the ability to address boosting, which is not available with the C4.5 classifier, and the C5.0 can deal with more complex data types than the C4.5 classifier. The improved performance of the C5.0 classifier can clearly be seen in the accuracy of the dataset investigated in this work. A set of parameters were used and the options of both classification training and testing were implemented. They collected accurate traffic data and then evaluated and compared the results obtained. Features were extracted mainly by taking the first flow of packets to build the ML dataset. The features were: number of send and receive packets; total

payload size (bytes) in the session; percentage of send to receive packets; payload size (bytes); ratio of small send packets of less than 50 bytes to all send packets; and the ratio of small packets received of less than 50 bytes to all received packets. They also used the number of acknowledgements and push flags added for the send and receive directions. The network traffic was classified by choosing seven different applications and using the C5.0 supervised ML technique. The results achieved 99% accuracy, although only for offline traffic and for a particular type of traffic. However, the complexity of the supervised ML and the parameters used in this work make it difficult to apply real-time network traffic classification.

García-Dorado et al. (2012) proposed a method to identify user activity and habits on the internet. They used DPI and statistical techniques to implement traffic classification and capabilities to characterise the activity of users and their habits in different countries. User behaviour was characterised and identified by generating traffic for the individual user while running a specific application. Users were identified when using an IP address and their traffic session information was collected. A calculation of the download and upload packet size was done for each user in relation to three types of application usage: streaming, file hosting and social networking. The total amount of download traffic data and session-downloaded bytes were analysed, in addition to the number of connections on each session used by the user. It was found that the user behaviour pattern was likely to change over different time scales, which created difficulty in the analysis of user traffic information. However, García-Dorado et al. (2012) did not choose more features to see the impact of other factors and view more user activity.

In their paper, Suznjevic et al. (2014) proposed a supervised ML (decision tree) algorithm to detect and classify user behaviour. Different behaviour classes were identified after the traffic analysis was applied to extract the classification parameters. Suznjevic et al. (2014)

suggested that this method could be used for profiling user behaviour. User behaviour classes were identified and categorised as idle if there was no activity for 10 seconds, and for browsing, audio or video streaming. Moreover, the parameters that were extracted mainly depended on uplink and downlink bandwidths. The mps, number of packets and mean bandwidth usage were calculated for both uplink and downlink bandwidths. The most relevant parameter is the downlink bandwidth number of packets, as mentioned in the results, and the accuracy for classifying user behaviour was 89.7 %.

Vinupaul et al. (2017) designed a method of identifying users that utilised four types of supervised ML algorithms for extracting and analysing parameters from network flow information. The parameters were mainly extracted from flow records and were divided into user parameters and host parameters. The user parameters were the number of IP destinations visited in a session; the site of interest, which is the maximum number of IP destinations visited in a session; and the different applications used by the user from the flow duration and the top destination port number among the flows for a whole session. The host parameters extracted from the flow records indicated in this study were the average value of the source port in the whole session flows; the unique source port number, which depends on the host being assigned as a server or a client; and the unique number in the packets. The four supervised ML algorithms used by Vinupaul et al. (2017) were two phases of k-means, random forest and C4.5. The flow records of 65 users were collected over seven days to build the datasets. The accuracy in this experiment achieved 83% when adding the host parameters extracted from the flow records to the user parameters. Moreover, adding timing information from flow records gave more accurate identification patterns relating to the user.

Hong et al. (2015) proposed a supervised ML classification algorithm using SVM learning. NetFlow traffic data were used to extract the traffic classification parameters

from flow-level traffic information. The SVM proposed was demonstrated to be 10 times faster than the eight previously proposed techniques found in the literature. In this paper, the traffic classification training was speeded up using an iterative technique and 12 features were extracted. Furthermore, the flow features extracted the source port, destination port, average packet size, client to server bytes per second, average server to client bytes per second, number of client to server packets, number of server to client packets, number of client to server bytes, number of server to client bytes, ratio of the number of client to server bytes, number of SYN flags and flow duration (end time - start time) (Hong et al., 2015). An overall accuracy of 99.1% was reached using a different type of SVM learning algorithm. However, adding the average of the parameters extracted from this study gave a higher accuracy for peer-to-peer applications used by the user of 80%.

The above-mentioned studies applied a set of traffic classification techniques that are flow- or packet-based, by extracting a set of statistical features and applying ML. However, most of the studies utilised basic statistical features, such as the number of packets, size of packets and number of connections (Zhang *et al.*, 2011; Malott and Chellappan, 2014), and the features extracted were used for different purposes, mostly to classify applications, which is not what the current study is looking to achieve. Indeed, this current research focused on the extraction of statistical features from the flow level to identify users based on network application traffic, which is a more challenging classification problem. The inter-arrival time was utilised by several studies, as explained above (Jin *et al.*, 2012; Suznjevic, Skorin-kapov and Humar, 2014), for different purposes; in the current research, the flow inter-arrival time was used for the analysis of the flow sessions, as explained in section 4.5. Previous studies (Banse, Herrmann and Federrath, 2012; McDowell, 2013) reported that using a DNS query is more accurate than using IP destination for filtering connections in network traffic analysis and, from that point, the

DNS query utilised in this project to filter applications to identify and profile users is based on application flow session timing resolution, as explained in section 4.4.3.

## **2.6 Literature on behavioural profiling**

This section presents a review of the literature on user behaviour profiling. It contains two sections: section 2.6.1 presents the usage of behavioural profiling in identifying the user, and section 2.6.2 presents user behavioural profiling in intrusion detection systems (IDSs).

Research in this area started in the 1990s and, since then, a variety of studies have examined behavioural profiling from different perspectives, such as identification and intrusion detection. Behaviour profiling can be used to verify a user based on his/her previous history; it then creates a user template, which can be used to decide whether or not this kind of activity belongs to a particular user. The sections below discuss the different systems that have utilised behavioural profiling.

### **2.6.1 User identification**

This section explains the use of behavioural profiling in identifying users. For example, Banse et al. (2012) proposed a method of identifying users through their web activity based on the websites they visit. In their study, the DNS queries were collected for a full day to be used as a training set for user identification and behaviour profiling based on flow-level statistical features, such as calculating the inter-arrival time for the timing window, as well as basic features such as the number and size of packets. The number of connections was also utilised to re-identify users based on connection weighting. A behaviour similarity method was used to manage the highest probability resembling more than one user. The results demonstrated 66.2% accuracy for the performance of the system overall in identifying users from a dataset of 2,100 users connected daily. The DNS queries showed that user behaviour could be associated with the same user's

behaviour on different days. The results show that the user behaviour patterns' accuracy changes slowly by holding up along gaps between the training set and the test set. The advantage of using the DNS compared with just using the IP destination is that web servers have multiple IP addresses hosted by a content distribution network. However, DNS caching may prevent DNS queries if the website is visited multiple times. Another problem with only using DNS queries is that the IP destination is not achieved on the DNS server.

Yang (2010) proposed a method for profiling users based on retrieving the most recent session to identify the user and the hostname was collected to construct a user profile. Lift-based profiling was used, which performed well with a small number of users. In addition, the extracted features utilised in this study initiated a user behavioural pattern, which included the number of sessions, overall user behaviour patterns, recover patterns for all users, and a measure of the weight between two profiles. The dataset utilised in this study was implemented by including the web browsing history of 50,000 users over a one-year period. The performance of the system indicates a promising degree of accuracy in the classification method of 90%. However, the author chose specific users and 10 different sliding windows to investigate analytical accuracy. The 10 users were also chosen from a large dataset, which would affect the study's reliability.

Herrmann et al. (2012) proposed a method of user identification and profiling based on a re-identification model gathered from web users' activity sessions to solve the problem of changing IP addresses. According to Herrmann et al. (2012), the website used by a user reflects, to some extent, his/her interests, habits and social networks. The URL of some pages might disclose the user's identity. Therefore, the user's records from past activities could be linked to a given user. A dataset of 28 users was collected for evaluation using the Naïve Bayes classifier and an attack was undertaken to observe the hostnames visited.

The results show that the framework was able to re-identify up to 50% of the users. In the study, the user behaviour profile was created using hostnames; however, there is an issue with this approach as the number of users might not be sufficient to investigate whether the administrator could correctly identify users based on the history of the websites they visited. Nevertheless, the problem attempting to be solved here might be helpful in the field of user profiling and identification based on network traffic application monitoring.

Kirchler et al. (2016) proposed a behavioural-based tracking system to monitor user activities on the internet over a long period of time, despite the changing IP addresses. DNS queries were used by collecting the network traffic of 3,862 users and applying unsupervised ML (K-means). Kirchler et al. (2016) argued that the supervised ML that had been used in previous research was time consuming in terms of training, which is difficult for observers. In their proposed method, an active user can be tracked based on accuracy, which linked 73% of the active user's session during a period of 56 days and the accuracy increased to 80% when the period was reduced to seven days. Users with the same activity usage sessions were assigned to one cluster, which meant that users in the same cluster could be identified based on their usage. A significant issue with an approach that involves multiple users is that they might share the same IP address, which could affect the system's performance. However, using DNS combined with IP destination may solve this problem of sharing the same IP address.

Finally, Alotibi et al. (2017) proposed a biometric behavioural profile approach based on features extracted from application-level interactions. The metadata network information was used to identify users in terms of the applications used and the actions taken by the users on a specific application. The interaction features utilised in this study were the start/end time of interaction, source port number, service IP, service port number, transmitted and received packets, and transmitted and received size of packets. Nine applications were used in this study to identify user activity from the application level.

The network traffic metadata were analysed in terms of identifying the interactions by using the TCP and UDP protocol signatures, which were mainly constructed from the connection packet size. The network traffic used in this study was collected from 27 users during a two-month period, with 62 GB of data. A neural network classification phase was used in this study with a nine-layer input and one output. The results of this experiment were 98.1%, 96.2% and 81.1% accuracy for the top three applications based on user interactions. This high proportion of accuracy was due to the set of features utilised in this research and the analysis, which was repeated 10 times to investigate the variance in the network traffic that was considered to create a user behaviour profile template. However, Alotibi et al. (2017) only considered user interaction based on the application level to identify users, which is limited to just identifying the user. In contrast, the results achieved in this thesis show that the proposed system not only identified users, but also represented their daily activity usage based on timing features, which is more challenging.

### **2.6.2 Intrusion detection systems**

This section explains the use of behavioural profiling in an IDS, which is analogous to anomaly detection. Ryan et al. (1998) proposed an intrusion detection method that could monitor unusual user activity by employing a neural network method. A behavioural profiling method was used in this study to identify each user by learning the prints left when the system was used. The system was divided into two components: the first was training the neural network to identify the user from the commands used during the day; the second component was the system administrator matching the normal user profile sessions when running the neural network at the end of the day. If a normal pattern does not match, an investigation into the system has to be applied. The method was implemented in a UNIX environment by keeping the executed command and forming a histogram of each user behaviour pattern to investigate the user profile. The system

involved 10 users with 96% accuracy in detecting anomalous behaviour, with a 7% false alarm rate. The high level of accuracy achieved was due to the set of features used in the research and the duration (nine months) considered in order to create a user behaviour profile template.

Oh and Lee (2003) proposed an anomaly detection technique that uses an unsupervised ML algorithm (clustering) to examine the normal user behaviour activities in a host. The features were divided into two areas: the frequent range of user activities and the infrequent range. The features were divided into two areas in terms of modelling the normal behaviour of a user for each cluster after clusters were identified by the proposed algorithm. Internal and external difference measures were also used to measure the normal behaviour distance difference and ratio difference. The log data of users were collected for two months and extracted by the basic security module. The false alarm rate of all the measures in the proposed method was found to be less than 10% and the detection rate was almost 100%. However, noise data can have a significant influence on the grouping of a cluster, such that it may be difficult to produce an accurate result in some cases.

Similarly, Park et al. (2010) proposed an intrusion detection technique that continually models the normal behaviour of a user over an audio data stream. A grid-based clustering algorithm was used to make an effective cluster without maintaining any data element of an audio data stream. The method was used to build a user profile and new user activities reflected a continuous cluster and a user profile at the same time. The features were divided into two areas: the frequent range of the user's activity and the infrequent range. Statistical methods were applied to the two areas to detect a user's normal behavioural activity. The profiling method was divided into two: an internal summary and an external summary. The internal summary contained the cluster properties and the external

summary contained the statistics of the noise data objects. In contrast, a grid-based clustering algorithm was employed with one million 10-dimensional elements to analyse the clustering method. The results show that the detection rate of the proposed method reached 100% and the false alarm rate was 12%. However, the false alarm rate was very high compared with that of the previous study, although the detection rate was almost the same.

The studies discussed above have shown the ability to profile a user's behavioural activity using unsupervised ML (clustering). The next section shows the statistical approach to profiling behavioural activity used in an IDS.

Yung Lee (2007) examined the monitoring of network usage patterns and detecting anomalies by monitoring users' network behaviour. The behavioural analysis method was employed in a dynamic local area network (LAN) environment, which is more accurate in an anomaly detection system. Network packets were used to identify users by connecting them to their network resources. User had to be identified before extracting the features that were utilised to collect the packets and build a behaviour profile. The features that were extracted to build the user behaviour profile were as follows: the number of network connections, duration of the connection, direction of the flow, frequency and ratio of valid network packets, volume of packets, and length and size of network packets. An unsupervised clustering algorithm was applied to the user behaviour profiles to group them into different elements depending on the usage of the network. The dataset contained 20 users to identify and build the user behaviour profile. This study used features extraction and the monitoring of user behaviour activity to create a unique profile for each user to be used in anomaly detection, which means this technique could be applied by using the number of connections and the direction of flow for user identification and profiling based on network application monitoring.

Similarly, Stiawan et al. (2010) proposed a method to detect network traffic threats based on the identification of an insider's habitual activity. Habitual activities were examined by collecting and classifying packets based on the regular expression of applications used by the insider user combined with the existing server activities log. The system was divided into three components after profiling the user activity: normal, suspicious and malicious. The following features were used in this study to identify normal activity that passed through the network and from any outside host: packets that had a low time to live (TTL) value, packets with the same source/destination port, packets with a private IP address, packets with invalid TCP flags, packets with a 0 port number, packets with strict source routing options, and packets that were too short. In addition, user activity was classified based on the applications used by the user into www, collaborative workspaces, download/upload, streaming video, remote login, remote Virtual Network Computing (VNC) and mail to profile the user's network traffic. The dataset was collected from a network of 200 users using Wireshark to collect data over a period of six days, with 2.3M packets analysed. The results show that the detection rate reached 95.55% and the false alarm rate was 8%. Although the features that were used here for profiling users from network packets are promising, adding features extracted from the session information, such as the duration of a session and idle time between connections, might give more accurate results.

## **2.7 Discussion**

Previous studies in the field of user behaviour profiling from generic network traffic information have proposed numerous methods and techniques for combatting and reducing the existing issues. A port-based technique is no longer sufficient because of the random port numbers used for different applications, as examined by Erman et al. (2006). A payload-based technique is the most powerful in the traffic classification field, but a problem occurs with encrypted data in that DPI cannot access it, as discovered by

PéterMegyesi et al. (2015). Within the user traffic profiling domain today, statistical analysis of flow- and packet-based techniques has been proposed to extract relevant parameters in terms of identifying a user profile (Vinupaul et al. 2017; Melnikov & Schönwälder 2010). However, a user behaviour profile has noisy traffic; for instance, when a user browses YouTube, the Google server will be connected, which makes the identification of an individual user profile more challenging, as investigated by McDowell (2013).

A number of studies have examined user behaviour profiling from different perspectives, such as intrusion detection (Oh & Lee 2003; Park et al. 2010; Deri and Stiawan 2012) and identification (Banse et al. 2012; Gu et al. 2015; Kirchler et al. 2016). The techniques were primarily utilised to verify a user by storing his/her previous activities and deciding whether the respective user is legitimate or illegitimate. The nature of behavioural profiling features has played a key role in creating an accurate user profile. In contrast, various pieces of research have been conducted to explore the possibility of applying user behavioural profiling to increase the level of security in computer networks. Indeed, the early studies in this field employed an anomaly-based detection system to determine and cluster abnormal behaviour (Ryan et al. 1998; Oh & Lee 2003) and some studies utilised normal behaviour and compared it with any 'abnormal' behaviour on the same network (Mukkamala, Janoski and Sung, 2002).

Different classifiers and clustering algorithms are addressed in the literature, such as SVM and neural networks. Neural network supervised ML has been employed with behavioural profiles in most of the studies, due to its ability to solve the non-linearity problem (Bivens *et al*, 2002; Alotibi *et al*, 2017). Gradient boosting is another classification technique used in the field, due to its ability to build a series of small decision trees based on the collected data, each tree attempting to correct errors from the previous stage (Zhang, 2008; Phan,

Sun and Tay, 2017). Furthermore, according to the various studies referred to in the literature review presented in this chapter, different areas and sources have been applied to create user behavioural profiles, such as application usage and number of connections by utilising DNS queries (Banse, Herrmann and Federrath, 2012).

Based on the discussion of the above-mentioned studies, analysing network traffic to identify users might be helpful for various reasons, as features can be extracted to build accurate user profiles. User behaviour profiling is an appropriate solution for addressing user behaviour that changes with an application over time in computer networks.

In addition, in terms of traffic analysis, many tools have been released to collect, analyse and monitor this information, some of which are open source and others are commercial. Therefore, NetFlow was used in this project by analysing network traffic to extract a relevant feature for user identification and behaviour profiling based on statistical techniques. The extracted features are based on application-level flow session timing resolution, as explained in section 4.5. Furthermore, DNS queries were utilised in the pre-process step to filter applications to apply the flow inter-arrival time to generate the flow sessions, as explained in section 4.4.3. As stated earlier, there is a need for a method to identify and profile users from their network traffic footprint, which should not only rely on identifying users, but also represent their 24-hour daily activity based on time bin features. Furthermore, to the best of the researcher's knowledge, the extracted timing features utilised and presented in this thesis have not been investigated before and so the achievement of this thesis should help ISPs to utilise user profiles to improve security and organisational network policy. To investigate how a user could be identified by his/her network traffic, new user identification and behaviour profiling based on an application-level system can be utilised to differentiate users. Therefore, the main focus of the proposed system is to identify and profile users by statistically describing their usage using a combination of timing features based on the session and the timing and flow of

DNS filtering, as DNS queries have been used in previous studies (Plonka and Barford, 2011; Banse, Herrmann and Federrath, 2012) to track users' flow connections. Similarly, this project has used DNS queries to filter the user flow connections based on grouping and filtering the applications used by the users. The concept of a user session has been utilised with different definitions and for different purposes in previous studies (Iváncsy and Juhász, 2007; Melnikov and Schönwälder, 2010; Kirchler *et al.*, 2016). In contrast, in this thesis, a user session is defined as a group of related flows characterised by a flow inter-arrival time (i.e., the time between two consecutive flows) that is lower than a predefined threshold. The flow inter-arrival time is denoted by  $\tau$  (where  $\tau =$  the start time of the second flow – the start time of the first flow) and shows that a new session starts when the flow inter-arrival time is more than the defined threshold. In addition, IP/MAC (media access control) address mapping criteria were used to separate and tag the daily flow network traffic related to each user by conducting a truth table based on an Address Resolution Protocol (ARP) timestamp similar to the mechanism used by Sinha, Mitchell and Medhi, (2003). To conclude, user behaviour profiling is an appropriate solution to changing user behaviour and applications over time in a computer network, using a statistical approach to extract new statistical features based on time bins. Therefore, it can be noted that each method has its strengths and limitations according to the different circumstances.

## **2.8 Conclusion**

As presented in this chapter, previous studies have reported methods from different perspectives for user identification, behaviour profiling, traffic classification and scoring a positive performance. Behaviour profiling has been utilised in different areas, such as identification and intrusion detection, in terms of comparing normal behaviour with an attack or abnormal behaviour that might affect the network. In addition, behaviour profiling has been used for identification by tracking user behaviour from interaction

usage patterns. It has also been shown that using accurate statistical features can contribute to creating an accurate template of a user profile. In addition, the outcomes discussed in the literature review reveal that user profiling and behaviour and network applications that change as the online interaction environment changes limit the existing literature with regard to the specifics of the topic of investigation.

Many techniques have been used in the area of behaviour profiling in the network environment. It also depends on the reason for profiling users, and most of the behaviour profiling research has investigated behaviour patterns based on normal usage as a basis for each user to be compared with any suspicious or abnormal activity that might affect the network sources. Behaviour profiling has been studied in terms of helping network administrators, ISPs and policy assessors to examine the information and make a formal decision regarding security, network monitoring, quality of service and managing bandwidth.

# **Chapter Three**

## **User behaviour profiling using an application-level flow sessions**

## **3 User behaviour profiling using an application-level flow sessions**

### **3.1 Introduction**

This chapter investigates the feasibility of classifying and identifying users' behaviour from network traffic metadata based on application-level flow analysis and statistically based timing features. As outlined in section 2.6, the existing research is limited with regard to user identification and behaviour profiling using application-level network traffic and has a number of associated challenges. One of the main problems is that relying solely on computer network internet protocols (IP addresses) to directly identify individuals who generate such traffic may not be reliable due to dynamic changes. Another issue is that the behaviour and underlying interactions of network applications are constantly changing, with more complex patterns being generated and, in parallel, user behaviour also varies and adapts as the online interaction environment changes. Thus, user identification and behaviour profiling in real-time network management remain a challenge. A challenge also lies in how to analyse generic network traffic metadata to describe a user's activity adequately and to identify the user and his/her behaviour over time. Furthermore, there are complexities around observing the traffic that belongs to a specific user due to traffic noise and the dynamic variation in the network resources, such as IPs, as mentioned earlier. Therefore, there is a significant interest in how to identify users and behaviour profiling from generic network traffic metadata for traffic engineering and security monitoring, as discussed in section 2.5. Policing, traffic management and investigating different network security perspectives, as well as attempting to detect anomalies in the way applications are used and highlighting these to a security entity, have increased the interest in identifying users and creating a user behaviour traffic profile to support network security administrators and ISPs in making informed decisions.

This chapter contains four main sections. Section 3.1 introduces the proposed user behaviour profiling using application-level flow sessions and the problem being investigated. Section 3.2 explains the details of the proposed behaviour profiling using an application-level flow sessions system. Section 3.3 discusses the proposed system as well as its mechanisms for user behaviour profiling, the proposed flow sessions timing resolution features and the research questions posed. Section 3.4 concludes the outline of the proposed model and the concept of using an application-level flow sessions system for user and behaviour profiling.

### **3.2 Flow inter-arrival timing, sessions and features**

In this research, user behaviour profiling using application-level flow sessions timing features had the aim of employing a range of mechanisms (IP/MAC address mapping, domain name query filtering criteria and the flow session concept) to overcome the issues referred to in section 3.1. To discriminate between the behaviour of different users, applications were grouped into sessions on data frames by analysing network traffic metadata and using DNS filtering criteria based on a web application. Predefined keywords were used to extract the relevant timing features from the start/end time of the application sessions related to each user. The concept of a session can be described as a group of related flows characterised by a flow inter-arrival time (i.e., the time between two consecutive flows) that is lower than a predefined threshold to divide the connections into sessions, with a start and end time for each session on a specific application. User behaviour profiling using an application-level flow sessions system led to developing a mechanism during the data collection for user traffic tagging based on a combination of timestamps, MAC/ IP address mapping, to produce a truth table based on the ARP while trying to avoid dynamic change. The proposed system employs DNS lookup criteria to deal automatically with user network traffic metadata, as one of the reasons for automating the analysis and filtering the criteria mechanism used by converting the IP

destination to the DNS was to enhance how traffic was dealt when related to users. DNS queries were used for filtering and grouping the applications requested by the user based on predefined keywords; for example, for 'bbc-vip016.cwwtf.bbc.co.uk', the predefined keyword is 'bbc.co.uk'. The domain name queries filtering criteria were allocated to filter applications and grouped them together to apply the session concept to the grouped web applications. The proposed system (IP/MAC address mapping, DNS query filtering criteria, and the flow session concept) led to a set of flow session activity timing features to discriminate users based on their interaction and daily usage. Furthermore, timing features for one hour (0-23) and quarter-hour (0-95) time bins were used to identify users and represent their 24-hour daily activity. For instance, one user might typically access Facebook at a certain time of the day (8 am to 9 am), whereas another user may access Facebook from 10 am to 11 am, which means the system could represent these different scenarios on a daily basis.

As discussed in section 2.5.4, statistically based approaches are the most reliable in the field of user identification and behaviour profiling, as they provide powerful mechanisms to differentiate and identify users based on their behavioural activity by extracting appropriate features. The aim of the system is to identify users' behaviour based on their networking activity footprint. Therefore, user network traffic is organised in flows and connections, which are organised into sessions to produce flow-level timing feature sets. As discussed in section 2.5, there is a notable lack of user identification and behaviour profiling based on application-level flow network traffic. Section 3.2.1 explains the definition of flows grouped into sessions as a concept proposed for the system to identify and profile users based on features that describe their network-related activities using time bins of a predefined resolution, which will be described here as activity timing features. Furthermore, the mechanisms utilised in the system were split and the traffic initialised (based on an IP source/MAC address source truth table and DNS queries)

because users' traffic needs to be tagged and the applications used also need to be tagged according to domain name filtering criteria. Section 3.2.2 explains the main concept of the proposed hourly timing features, which are extracted based on the flows grouped (connections) into sessions for user identification and behaviour profiling. This is based on the application-level network traffic footprint to identify and represent the daily usage of a user by dividing the day into hours (0-23) and quarter hours (0-95). Section 3.2.3 explains the flow session extracted statistical features that have been proposed by previous studies and used in this project based on the basic NetFlow tuples.

### **3.2.1 Flow session definition and traffic splitting**

The main focus of the proposed system is to identify and profile users by statistically describing their usage through a combination of features based on the session, timing and flow DNS filtering. A user session can be described as a group of related flows characterised by a flow inter-arrival time (i.e., the time between two consecutive flows) that is lower than a predefined threshold. The flow's inter-arrival time is denoted by  $t$  (where  $t = \text{the start time of the second flow} - \text{the start time of the first flow}$ ), and a new session starts when the flow inter-arrival time is above the defined threshold. Accordingly, a session is a way of representing a user's behavioural activity and changes based on timing feature bins extracted from the start/end time of each application session (hour session timing features and quarter-hour session timing features). The assumption is that a user browses several applications in different timing slots, which requires the extraction of timing features to identify and represent that user's behaviour. The timing features represent the user's behavioural activity based on the applications used by the user through the number of connections, type of application and the exact day and time that the application is browsed. The following example explains the concept of session timing resolution and how it can be used to identify the behaviour of the user. A user is browsing an application, such as Facebook ([www.facebook.com](http://www.facebook.com)) or Instagram

(www.instagram.com), on Saturday: if Facebook is browsed from 6 am until 3 pm and Instagram from 4 pm until 10 pm, and another user browses Facebook from 11 pm until 12 pm and Instagram from 1 am until 3 am, these timing slots would be represented in a daily 24-hour timing space based on the proposed session concept. Therefore, user identification and behaviour profiling are needed to investigate each user in order to build an individual profile of his/her daily activity, as well as to investigate the applications browsed during the day based on the flow session timing resolution concept to overcome the problems described in section 3.1.

To investigate the proposed session concept for user identification and the behaviour profiling system described in section 3.2, the actual network traffic (NetFlow) was collected and pre-processed and then separated based on IP/MAC using an ARP mapping truth table. The truth table was used to match the user IP source with the source MAC address and a timestamp using a similar approach to some of the previous studies (Kabir, Mudur and Shiri, 2012; Algiriyage, Jayasena and Dias, 2015). Therefore, separating users by relying only on IP addresses can provide reliable results for only a short time - minutes or hours - but is not reliable if actual data are collected for a long time, such as over months. The rationale behind the implementation of the IP/MAC address mapping method is that the data were to be collected over a long period of time, which means there is a possibility that the users' IP addresses would alter during that time due to the dynamic change caused by the DHCP. In addition, the applications requested by a user were identified using the DNS queries lookup method. The method was used to convert the IP destination to the DNS to enhance the way the traffic related to each user is dealt with, and to filter and group the applications requested by the user based on predefined keywords; for example, for 'bbc-vip016.cwwtf.bbc.co.uk', the predefined keyword for this web application is 'bbc.co.uk'. This approach is similar to that utilised in some previous studies (Banse, Herrmann and Federrath, 2012; McDowell, 2013). Consequently,

the proposed session concept approaches user behaviour profiling by using application-level flow session timing resolution features, and leads to a set of features that represent user activity behaviour divided into a 24-hour daily timing space by utilising two time slot resolutions (hour and quarter hour). In contrast, a number of studies have examined user behaviour profiling from different perspectives, such as identification to distinguish users (Melnikov and Schönwälder, 2010). The techniques employed are primarily used to identify a user by storing previous user activities in order to decide whether the user being examined is legitimate. Therefore, a number of studies (Haiyan *et al.*, 2007; Alotibi *et al.*, 2017) have used behavioural- and statistical-based techniques by observing the interaction of the client with network applications and proposing a set of features based on connection timing for user identification and profiling, such as the average packet size while uploading a video on YouTube. Another group of studies (Lim and Jones, 2008; Park, Oh and Lee, 2010; Paredes-Oliva *et al.*, 2012) explored the possibility of applying user behavioural profiling to increase the level of security in computer networks. Indeed, early studies in this field employed anomaly-based detection to determine abnormal behaviour. It can be asserted that using behavioural profiling can help in differentiating users for various purposes and different performance measures based on the statistical features extracted from generic network traffic and different activities could be provided to build an accurate user profile. Another study (Iváncsy and Juhász, 2007) proposed a method for identifying web users based on the log files from the user session. In that study, a session is defined as a sequence of activities performed by a user when browsing a specific website. The proposed session concept is split into two daily timing slot resolutions: the first time slot resolution is one hour, and the second time slot resolution is quarter of an hour.

### 3.2.2 Flow session timing resolution features

As referred to above, two time slot resolutions (hour and quarter hour) are proposed to identify users based on an application-level flow session to describe 24-hour daily usage activity, which is extracted based on the flow session defined in section 3.2.1. In addition, the statistical features (max, min, mean and median) extracted from the flow session proposed in previous studies were applied to the proposed flow session timing resolution to assess the effect of these features on the accuracy of the profiling, as explained in section 3.2.

Examples of daily 24-hour activity based on flow session timing resolution features are illustrated in Table 3.1 and Table 3.2. The daily 24-hour usage timing resolution and activity might vary between users because, for example, User  $x_i$ 's interaction activity will differ from that of User  $y_i$ . Therefore, two types of flow session timing resolution activities are presented in this project to identify and differentiate between user behaviour and activity during a 24-hour slot, not only to distinguish the timing of the activity, but also the exact application and the day it occurred. First, the hourly flow session timing resolution is represented based on the start/end time of sessions using Python script and encoded as an integer input (0-23) timing resolution, to represent the daily usage of the user and enable investigation of the variance between users' activity. The hourly flow session timing resolution is encoded in terms of combining the start and end timing resolution for the whole session related to one application used by a user. Table 3.1 summarises the hourly timing resolution features.

In addition, the names of the 11 popular web applications selected (i.e., Amazon, Google, Instagram, Facebook, LinkedIn, Yahoo, MSN, Unknown, Stack Overflow, TeamViewer and IEEE) were encoded into an integer input (from 0-10), due to the possibility of user activity varying with the type of application with which the user interacts. Each

application needs to be indicated by an integer number to be applicable for ML, which requires input to be numeric rather than a label. Given that, activities may also vary with the date of the week, which is encoded as an integer input (from 0-6) to represent the exact day of the usage. The number of connections in each session, which might give discriminative information regarding the variation in the usage activity between users, were also utilised as input features. The hourly timing resolution features were encoded into 0 or 1 to represent the activity of a user in a specific bin. The concept of this binary notation is to assign 1 if the respective activity occurred during the time bin or 0 if there was no evidence of it during the time bin, as shown in Table 3.1, to represent a user's daily usage behaviour.

The above technique allowed activities to be summarised as each sample includes one application per user per day, which is extracted based on the sessions to be fed to the classifier. Therefore, the full list of hourly session timing resolution features represented in Table 3.1 includes the extracted statistical and hourly session timing resolution features. Therefore, the extracted statistical features represent the statistics (max, min, mean and median) of features that have been proposed by other works to investigate their influence on identifying and profiling users. The hourly session timing resolution features were proposed to represent the overall daily activities (day, start hour, end hour), the amount of data exchanged (no of connections) and application type (app), as well as binary application usage for all time slots (0-23). The following example explains the concept of the hourly flow session timing resolution features and how it can be used to identify the behaviour of a user. For instance, Table 3.1 shows that user  $x_i$  and his/her associated activity can be described as (web application Amazon (0), on Monday (0), time slots (0-5), with activity-coded values indicating his/her behaviour. Therefore, all these hour bins (time slots) will be 1s if seen in the slot and other bins will take 0s if not seen in the slot, or if user  $x_i$  performed the activity during the respective slot. Another usage activity for

user  $x_i$  could be represented as web application Facebook (1), on Monday (0), time slot 2-4; this hour bin will take 1s and the other bins will take 0s if user  $x_i$  performed the activity during the respective slot. If user  $x_i$  uses web application Instagram (2), on Monday (0), time slot 1-3, this hour bin will take 1s and the other bins will take 0s if the activity is performed in the respective slot. If user  $x_i$  uses web application Yahoo (3), on Monday (0), time slot 21-23, this hour bin will take 1s and the other bins will take 0s if the activity is performed during the exact slot as the input matrix, as illustrated in Table 3.1. The input matrix shows the day and its encoded values to represent the daily 24-hour user activity based on the hourly timing resolution representation. The proposed flow session extracted statistical features, and hourly flow session timing resolution features were fed to the gradient boosting decision tree classifier to be evaluated as one sample in the investigated dataset, which represents one application per user per day, with 101 features extracted. Each row represents an independent sample, which is processed by the ML classifier independently as an intra-class data point; the full list is shown in Table 3.3.

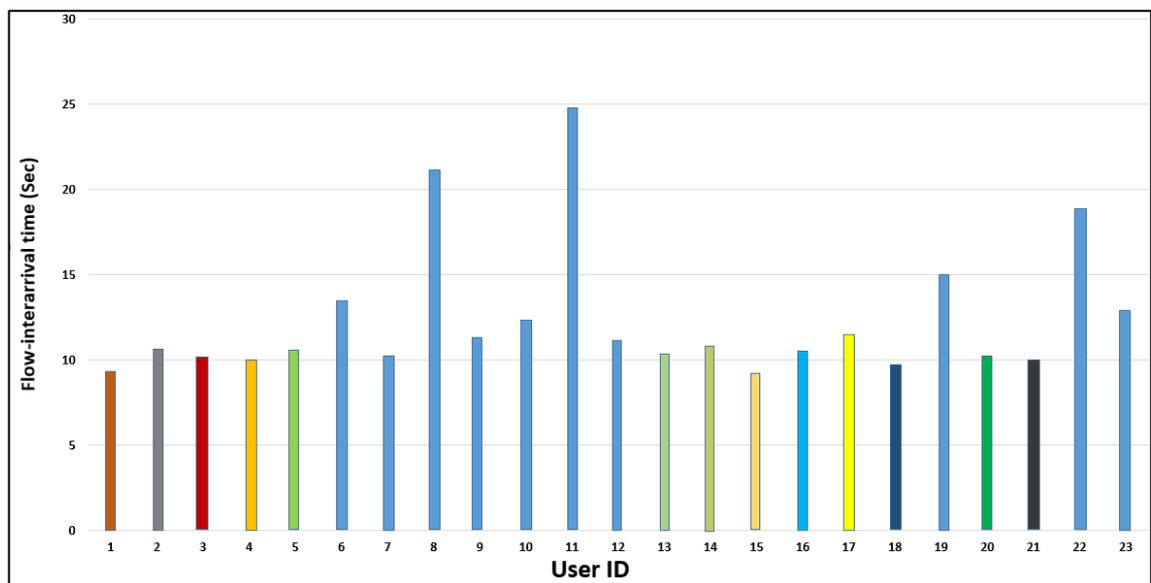
Generally, with respect to the particular features extracted at the application level, the session timings (24 one-hour and/or 96 quarter-hour bins) play a big role in user identification. Therefore, the timing features enhanced the way in which users can be identified, as hourly investigation differs from quarter-hourly examination. For example, some users used their application in different timing slots: some used Facebook every hour and others used it every quarter hour. As a result, extracting novel features are the most important point to enhance in any identification and behaviour profiling system, as the quality of the system depends upon the quality of the extracted features.

**Table 3.1: Hourly timing resolution**

No/samples	User	Extracted Statistical Features				Hourly Session Timing Resolution Features														
		Max in Bytes	Min in Bytes	Mean in Bytes	Median in Bytes	No. of connections	App	Day	Start hour	End hour	0	1	2	3	4	5	..	21	22	23
1	1	476	567	890	345	141	0	0	0	5	1	1	1	1	1	1	..	0	0	0
2	1	679	897	546	904	66	1	0	2	4	0	0	1	1	1	0	..	0	0	0
3	1	123	456	345	724	80	2	0	1	3	0	1	1	1	0	0	..	0	0	0
4	1	336	612	591	810	70	3	0	2 1	2 3	0	0	0	0	0	0	..	1	1	1
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
5	23	567	789	345	562	66	0	3	0	1	1	1	0	0	0	0	..	0	0	0
6	23	782	882	629	456	78	1	3	2	3	0	0	1	1	0	0	..	0	0	0
7	23	435	561	788	345	98	2	3	3	5	0	0	0	1	1	1	..			
8	23	345	346	908	276	345	3	3	2 1	2 3	0	0	0	0	0	0	..	1	1	1

The user identification and behaviour profiling based on the session concept is defined as identifying and linking the user session based on DNS criteria, as discussed in section 2.6.1. A previous study (Vinupaul *et al.*, 2017) proposed different session flow inter-arrival time thresholds of 3 sec, 6 sec and 12 sec and changing between these threshold values based on the different environments investigated. Another study (Oudah *et al.*, 2019) proposed different statistical values for the inter-arrival time threshold of 10 sec, which suggests that the value of the threshold could be assigned as any static or dynamic value depending on the definition of the session and any distribution analysis to investigate the nature of the data, which are sometimes affected by the environment. Figure 3.1 shows the distribution of the flow inter-arrival time threshold based on the preliminary analysis, where the X-axis is user ID and the Y-axis is the flow inter-arrival time frequency. However, the distribution analysis and investigation of the dataset

showed that most of the distribution frequencies fell at around 10 sec, as shown by the frequency for many of the users (1, 2, 3, 4, 5, 13, 14, 15, 16, 17, 18, 20 and 21), except for those users (6, 7, 8, 9, 10, 11, 12, 19, 22 and 23) for whom the frequency falls below 10 sec. Therefore, the hourly timing resolution flow inter-arrival time threshold was assigned as 10 sec based on the majority of the users' frequencies.



**Figure 3.1: Distribution of the hourly flow inter-arrival time threshold**

Second, the quarter-hour flow session timing features, which were based on the same concept, were applied to the hourly flow session timing resolution using the session concept discussed in section 3.2. The main idea behind the quarter-hour timing resolution features was to identify a user and represent his/her 24-hour daily activity based on shorter timing slots, instead of larger one-hour time slots. The quarter-hour data were extracted to compare them with the hourly timing-based resolution to assess the influence of the different time resolutions implemented and presented in this section (3.2.2). The calculation was applied at the start/end time of the session to divide the daily usage of the user into quarter hours to represent his/her daily usage based on the timing resolution bins based on the sessions and was encoded as an integer input (0-95) using the equation below. The quarter-hour timing resolution features were encoded into binary time bins (0 or 1)

to identify the user and to represent the user's exact quarter hour time slot, also using the following equation:

$$\text{quarter of hour bins} = (60 * 60 * 24) / (60 * 15) = 96$$

The following example explains the concept of quarter-hour flow session timing resolution features and how they can be used to identify a user's behaviour. For instance, user  $x_i$  and his/her associated daily activity could be described based on a quarter-hour timing resolution as (web application amazon (0), on Monday (0), time slot quarter 0-3) per day; all these quarter-hour bins will take 1s and the other bins will take 0s to represent the user's activity and his/her behaviour. The encoded matrix gives an indication of the exact usage of Amazon on Monday, as Amazon was used between quarters 0 and 15 if the activity is performed in the associated time slot. User  $x_i$  and his/her associated activity could be represented based on quarter-hour timing resolutions as web application Facebook (1), on Monday (0), time slot quarters 1-4; this quarter-hour bin will take 1s and the other bins will take 0s if user  $x_i$  performs the activity during the respective quarters. If user  $x_i$  uses web application Instagram (2), on Monday (0), time slot quarters 93-95, this quarter-hour bin will take 1s and the other bins will take 0s if the activity is performed during the respective slot. If user  $x_i$  uses web application Yahoo (3) on Monday (0), time slot quarters 2-3, this quarter-hour bin will take 1s and the other bins will take 0s if the activity is performed during the exact slot. To further represent the details of the session concept, in addition to the timing features, the number of connections for each session is calculated to investigate the daily amount of data exchanged during the usage. The number of connections is extracted to represent the user activity, not only based on the timing features and their encoding, but also to represent the number of connections for each session predicted, as illustrated in Table 3.1 and Table 3.2.

Previous studies (Sinha, Mitchell and Medhi, 2003; Garsva et al, 2014) have proposed different ranges for the inter-arrival time threshold, which was chosen as 5 sec, 10 sec and 15 sec to produce different datasets; each dataset was assigned one of the randomly assigned thresholds. The reason for producing different datasets, each with a different threshold, was to analyse and compare the different threshold values to assign the user session and identify the web user based on the log file. Therefore, similar to the earlier work in this study, the flow inter-arrival time threshold for the quarter-hour timing slot features experiment was assigned to different values for the different datasets. Given that, during the quarter-hour timing features experiment, the threshold could be set differently (to 6, 12, 18, 24, 30 or 36 sec), which differs from previous studies in this field. In previous studies, web users were identified based on the log file with a different definition of the session. In the current study, different thresholds were set for the sessions implemented for the quarter-hour timing resolution features (6, 12, 18, 24, 30 and 36 sec). Assigning different thresholds was done to investigate the most visible quarter-hour daily timing slot resolution features and, as found in the analysis and experiment, the dataset with 30 sec scored the highest accuracy of all of the datasets implemented with the proposed quarter-hour timing resolution feature used to identify users' daily activity. The proposed flow session extracted statistical features and quarter-hour flow session timing resolution features were fed into the gradient boosting decision tree classifier to be evaluated. One sample represents an application per user per day, with 172 features extracted to build the dataset investigated. Therefore, as illustrated in Table 3.2, each row represents an independent sample, which is processed by the ML classifier independently as an intra-class data point. The samples contain the extracted statistical features (max, min, mean and median) and the quarter-hour session timing resolution features proposed by the system.

**Table 3.2: Quarter-hour timing resolution**

No/samples	User	Extracted Statistical Features				Quarter-hour Session Timing Resolution Features														
		Max in Bytes	Min in Bytes	Mean in Bytes	Median in Bytes	No of connections	App	Day	Start quarter	End quarter	0	1	2	3	4	5	..	93	94	95
1	1	457	543	651	69	141	0	0	0	3	1	1	1	1	0	0	..	0	0	0
2	1	789	112	890	76	66	1	0	1	4	0	1	1	1	1	0	..	0	0	0
3	1	982	178	543	43	80	2	0	93	95	0	1	1	1	0	0	..	1	1	1
4	1	231	234	347	63	70	3	0	2	3	0	0	1	1	0	0	..	0	0	0
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
5	23	345	564	783	52	100	0	3	0	1	1	1	0	0	0	0	..	0	0	0
6	23	879	654	789	89	35	1	3	2	4	0	0	1	1	1	0	..	0	0	0
7	23	345	231	432	43	150	2	3	4	5	0	0	0	0	1	1	..	0	0	0
8	23	221	456	184	67	120	3	3	93	94	0	0	0	0	0	0	..	1	1	0

### 3.2.3 Flow session extracted statistical features

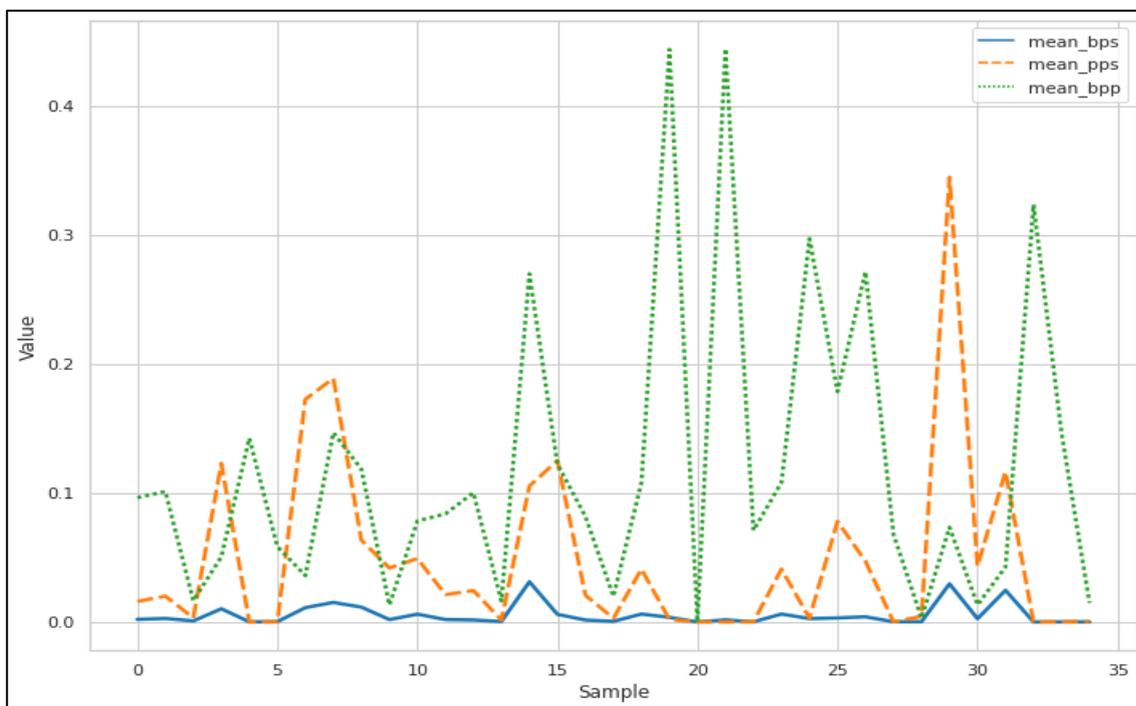
The statistical features extracted are part of the proposed flow session timing resolution concept and the timing resolution features from which the full statistical set (min, max, mean, median, etc) are extracted for each time slot resolution feature (hour and quarter hour). The statistical set (min, max, mean, median, etc) has been provided in order to summarise the distribution and to verify which of them impacts significantly on the accuracy of the profiling. Some of the extracted statistical features have been proposed by previous studies, as discussed in 2.5.4, and are the output from the basic NetFlow (using nfdump) (Haag, 2006) tuples, such as the following: In pkt, In byte, Out pkt, Out byte, bit per second (bps), packets per second (pps) and bytes per packet (bpp). Therefore, to explore the prediction ability of NetFlow attributes using the proposed methodology, a calculation was made for each flow session based on the basic NetFlow tuple features selected to be part of the proposed flow session timing features, as well as obtaining

statistical data from these features based on the equations presented in Table 3.3. The reason for calculating the extracted statistical features was to enhance the timing resolution features and to observe the influence of these features on the accuracy of the profiling. The statistics also vary for the frequencies in terms of the discrimination of the statistical number, as the max, min, mean, median, etc. of these features might differ from user to user depending on the application used and the time spent browsing the application. In contrast, the other features (d2s bps, d2s pps, d2s bpp, transmitted data rate, received data rate, received to transmitted packets, received to transmitted data) were calculated based on equations from the basic tuples (In pkt, In byte, Out pkt, Out byte, bps, pps, bpp) and the duration of the session in seconds, as these types of feature do not output directly from NetFlow (nfdump), as illustrated in Table 3.3.

**Table 3.3: Flow session timing activity features**

<b>Features</b>	<b>Description</b>
User ID	User 1, User 2, User n
<b>Flow session extracted statistical features</b>	
In pkt	session source (client) to destination (server) packet
In byte	session source (client) to destination (server) byte
Out pkt	session destination (server) to the source (client) packet
Out byte	session destination (server) to source (client) byte
bps	session source (client) to destination (server) bits per second
pps	session source (client) to destination (server) packets per second
bpp	session source (client) to destination (server) bytes per packet
d2s bps	session destination (server) to source (client) bits per second (8* out byte /duration(sec))
d2s pps	session destination (server) to source (client) packets per second (out packets/duration(sec))
d2s bpp	session destination (server) to source (client) bytes per packet (out bytes/ out packets)
Transmitted_data rate	session source (client) to destination (server) data rate (in byte/duration (sec))
Received_data rate	session destination (server) to source (client) data rate (out byte/duration (sec))
Received to transmitted packets	session received to transmitted packets (out packets / in packets)
Received to transmitted data	session received to transmitted (out byte / in byte)
<b>Flow session timing resolution</b>	
Number_of_connections	Session number of connections
Day_of_the_week	Date encoded (0-6)
Application	Application name encoded (0-10)
Start_hour / quarter	Integer encoded hourly (0-23) and quarter hour (0-95)
End_hour / quarter	Integer encoded hourly (0-23) and quarter hour (0-95)
Application_usage	Start / end hour integer hourly (0-23) and quarter hour (0-95) represented (0 or 1) in time bins

Figure 3.2 shows the value distribution of the flow session extracted statistical features (mean bps, mean pps and mean bpp) of the subset samples to demonstrate the difference between the features. The figure shows that the mean bps, pps and bpp belonging to the 35 samples taken from the dataset have different values, in addition to the variance in their calculation equations shown in Table 3.3. The variance in the values was used to enhance the proposed timing features and increased the ability of the system to identify and profile users based on the unique statistical information added (max, min, mean, median). This enhanced the ability of the classifier to differentiate users. The flow session extracted statistical features were selected based on their reasonable performance in previous works related to the proposed system. The selection of important features was also conducted based on the feature analysis presented in section 5.2.4, which indicates the importance of the features and the variance in the values of the extracted statistical features.



**Figure 3.2: Distribution of the flow session extracted statistical features**

### **3.3 Discussion**

The proposed system for user identification and behaviour profiling from generic network traffic could be a reliable solution to the problems discussed in section 3.1. The set of features proposed is based on application-level flow sessions (inter-arrival time between flows) and DNS query filtering, in addition to IP/MAC address mapping. Therefore, obtaining precise user statistical features and session timing resolution patterns should lead to an accurate model for use in identifying users. The observation and calculation of user behaviour activity among different applications also provide a strong approach to identifying relevant traffic. When combined with DNS queries, user MAC address mapping and session timing resolution might provide a discriminative approach to targeting traffic. The analysis in this research relies on MAC addresses instead of IP addresses to ensure host consistency and the correctness of the user's data, as the DHCP changes users' IP addresses. As the proposed approach is based on timing resolution features, this might enhance the ability of the system not only to identify and profile users, but also to investigate the particular application browsed by the user, as well as the time slot of the browsing. Hence, the proposed flow session timing resolution features based on binary encoding could be used to represent the exact time of a user's activity on the application being investigated. In contrast, the encoding applied to the proposed system involves flow session timing features (hour and quarter-hour timing resolution), as well as the applications and days of the week; the encoding might make the data compatible with the required input for the processing i.e., numerical rather than categorical.

#### **3.3.1 Research questions identified**

Having introduced an approach to user profiling based on network application traffic, this research needed to investigate further how efficient and robust the proposed approach would be in profiling network users based on the traffic generated and the features set

identified. The following research questions were generated and are addressed in the following chapters:

1. How identifiable and unique is user behaviour based on the network traffic generated?
2. How variable is user behaviour over time in the context of the dynamic and changing nature of network applications?
3. What is the effect when using different time resolutions for profiling users during the processing of the collected data (hours and quarter hours)?

To answer the above research questions, a set of experiments and analysis were conducted to identify the strengths and limitations of the proposed approach. The experiments and their analyses are presented in Chapters Four, Five and Six.

### **3.4 Conclusion**

This chapter explored a novel system that is proposed in order to identify and profile users based on application-level flow session timing resolution bins identified based on flow inter-arrival time using IP/MAC address mapping and DNS filtering using network traffic metadata. The proposed system builds on research identifying unique user behaviour, as the behaviour and underlying interactions of network applications are constantly changing. In parallel, user behaviour is also changing and adapting. The proposed system is based on timing resolution features that give the system the ability to investigate the particular application browsed by a user, as well as the time slot and the duration spent browsing that application. To this end, user behaviour profiling was deemed an appropriate solution for exploring the changes in user behaviour and application usage over time in a computer network by using statistical approaches that involve extracting new statistical features based on timing criteria. The next chapter explains in detail the general block diagram (data processing pipeline) of the user behaviour profiling of the

proposed application-level flow sessions system, together with the components and analysis of the mechanisms implemented.

# **Chapter Four**

## **Methodology and data collection**

## **4 Methodology and data collection**

### **4.1 Introduction**

The previous chapter outlined the concept behind the proposed system, including definitions of the problem and of a flow session, as well as the application-level timing features extracted for user identification and behaviour profiling. This chapter explains the general block diagram of the proposed system, from data collection to classifier implementation, and evaluates the comprehensive features extracted based on the application-level flow sessions identified from DNS filtering criteria and flow session timing features. The collected NetFlow data (nfcapd files) are analysed using the nfdump analysis tool to gain the basic tuples. The IP/MAC address mapping mechanism was implemented in the system to eliminate reliance on IP addresses, as these are changeable and unreliable, and is based on ARP mapping, as explained in section 3.2.1. The DNS lookup system was used to initialise the application being used by the user and to filter the applications using predefined keywords to apply the concept of the flow session. The pre-processing continued until a way of extracting the novel daily hour and quarter-hour flow session timing features explained in section 3.2.2 were reached. Section 4.2 introduces the general block diagram of the proposed system. Section 4.3 explains in detail the data collection mechanism and the procedure conducted to take the user's privacy permission to collect his/her NetFlow traffic data. Section 4.4 presents the user behaviour profiling using application-level flow session system data pre-processing mechanisms from the traffic dump by using the nfdump tool, IP source and MAC mapping, DNS lookup process, application flow filtering mechanism and flow session generator, which uses Python script to apply the concept of the proposed flow session. Section 4.5 discusses the extracted features, as well as the input data, and builds upon the proposed feature sets. Section 4.6 explains the classification step used to feed the set of extracted statistical features and timing resolution features proposed to identify the user,

using application-level network traffic for the classification, in order to evaluate the proposed system's timing features. Section 4.7 provides a discussion of the general block diagram in relation to user behaviour profiling using an application-level flow sessions system, including the mechanisms from the data collection for extracting the system's novel timing features. Section 4.8 concludes the chapter on the experimental methodology and data collection for user behaviour profiling based on an application-level flow sessions system.

## **4.2 General block diagram for the proposed system**

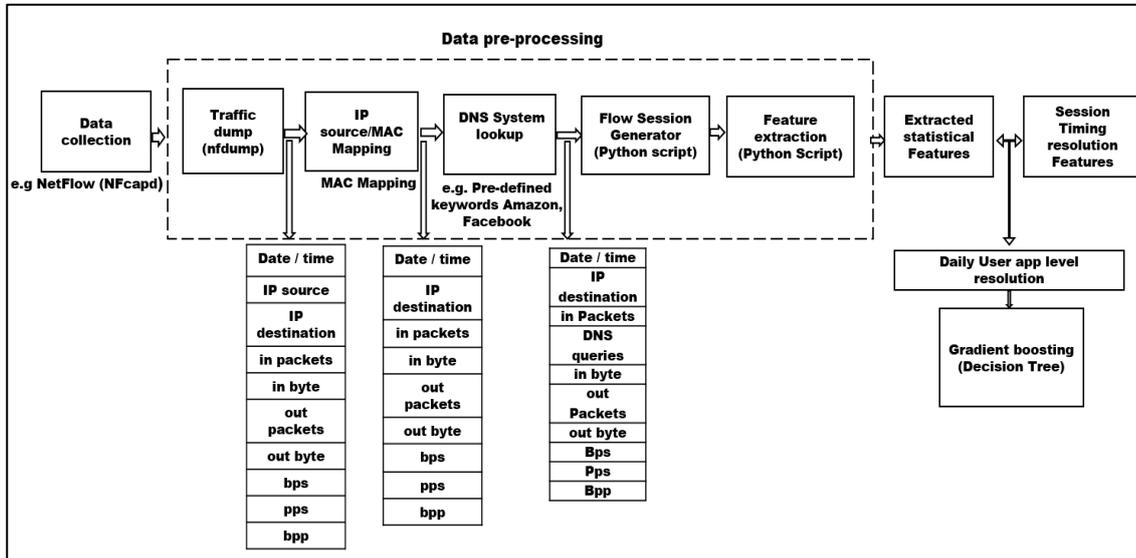
The method followed in this research focuses on extracting and analysing a flow-level features set that allows identification of user behaviour through its network activity footprint based on flow session timing resolution features and extracted statistical features. A set of features was used to investigate users' identification and their daily internet usage activity based on a filtered applications session, as explained in section 3.2.1. The applications involved were identified based on DNS query lookups (Internet Systems Consortium, 2005) by converting the IP destination (number) to a DNS query (text) to initialise the application being used by the user and filter the applications based on predefined keywords to be implemented in the flow session concept pre-process, as discussed in section 3.2.1. The concept of a user flow session proposed in this research represents the daily usage of the user based on the applications used and timing resolution features, as well as the statistical features extracted, which not only identify the user, but also represent the exact time slot the application is used, as discussed in section 3.2.

The threshold value was determined by conducting a preliminary analysis to assign a value for the flows in both the hour and quarter-hour timing slot resolutions; the threshold was assigned different values based on the trialling of different datasets, as discussed in section 3.2.2. Using session characteristics as a discriminator is based on user behaviour

differing between users (for instance, browsing Facebook varies from user to user in timing and content), as discussed in section 3.2.1. Accordingly, a session is a method that might help discriminate users' behaviour and their 24-hour daily activities based on time slot resolution bins extracted from the start/end time of each application session (using hour and quarter-hour resolutions). The top 11 applications were selected based on a statistical procedure, computed by implementing the DNS query keywords for all users in order to count the connections for each application, and then choosing the most used applications and websites (i.e., Amazon, Google, Instagram, Facebook, LinkedIn, Yahoo, MSN, Unknown, Stack Overflow, TeamViewer, and IEEE). These applications were added to the session generator for application filtering and labelling purposes. Users were filtered using MAC address mapping to label the data related to each use.

To validate the above method, an experiment was carried out using a dataset captured from the University of Plymouth, Centre for Security Communications and Network Research (CSCAN) laboratory for 23 users. The raw network traffic was stored as NetFlow records using nfdump (Haag, 2006). The stored flows were processed using Bash and Python scripts to filter users based on IP/MAC address mapping and applications based on DNS queries to extract statistical features and flow session timing resolution features. The proposed timing resolution features and extracted statistical features were summarised statistically to produce daily user application-level records that were analysed based on the Z-score normalisation process to obtain features on the same scale, similar to previous studies discussed in section 2.5.4. Then, the dataset was fed to a random forest algorithm to explore feature importance based on a ranking mechanism. After the dataset had been normalised and the feature importance mechanism applied, a gradient boosting decision tree classifier was used to feed the dataset into the classifier to evaluate the proposed flow session timing resolution features (hour and quarter-hour timing features) and the extracted statistical features, as explained in section 3.2. Figure

4.1 shows the experimental methodology and the steps taken, from data collection to the classification process, to evaluate the extracted statistical features and timing resolution features proposed for the system. More details are provided in the following subsections.



**Figure 4.1: Framework of proposed user behaviour profiling using an application-level flow sessions**

### 4.3 Data collection

The dataset was collected for 23 users over a period of 60 days (from 8 May until 8 July 2018), following ethical approval from the University of Plymouth Research Ethics Committee (see Appendix). All users were connected to the same local network within CSCAN at the University of Plymouth. Of the 23 users, 20 were PhD students and three were master’s students, all of whom were working on their research at the laboratory. The 20 PhD students were aged between 32 and 45 and the three master’s students were between the ages of 25 and 30. Thus, all 23 users were between the ages of 25 and 45, which is a limitation of the research as the results may not be extendable to users under 25 or over 45. In terms of gender, 10 of the users are female and 13 are male, which will not affect the performance of the system as most of the users were working during the day with their nature browsing while the data collected during the 60 days.

As the aim of this research is user identification and behaviour profiling, utilising NetFlow records rather than packets is more appropriate for the following reasons. First, NetFlow records enable greater accuracy in identifying the web applications used. Second, NetFlow records focus only on connection metadata, ignoring the transmitted packet data and payloads. Third, collecting NetFlow data enables less data storage in comparison with collecting each transmitted packet. The nfcapd tool was used as a collector via a network-based method, as explained in section 2.3.1. During this period, the participants accessed the internet through the university network. Participants were not asked to follow a protocol but to use their device(s) in their typical fashion. The data were collected during the participants' browsing of the internet and stored in NetFlow (nfcapd file format) to be dumped in the next phase to extract the appropriate format for the pre-processing step and to extract the proposed features mentioned in section 3.2.2. The collected flow network traffic metadata, which were saved in nfcapd format (nfcapd.201806190002), produce a file every 5 min, resulting in 288 nfcapd files collected per day for all users and a total of 17,280 files collected for the whole 60-day period for all users. The following example represents the nfcapd files for one day: nfcapd.201806190002 and nfcapd.201806192357. The first (nfcapd.201806190002) represents the start of day 19/06/2018 from 12:02 am, and the second (nfcapd.201806192357) represents the end of day 19/06/2018 at 11:57 pm; the next day is represented as nfcapd.201806200002: nfcapd.201806202357, which is suitable as input into the nfdump tool. The basic NetFlow data are explained in section 2.3.1.

#### **4.4 Data pre-processing**

The data collected were pre-processed by generating bidirectional flow-based network traffic information. The network traffic was processed and generated in several steps to attain the most relevant flow-level timing features to identify users based on application sessions and timing features. This was achieved by ensuring that the flows correlated

when using DNS queries and application filtering to reconstruct the sessions. The following subsections explain the steps undertaken to pre-process the raw network traffic to extract the desired features.

#### **4.4.1 Traffic dump**

The network traffic data collected were analysed using the nfdump tool and its command line on Linux Ubuntu 15.10 by taking the nfcapd.201806190002: nfcapd.201806192357 format explained in section 4.3 as an input to dump the network traffic for all users. This process was applied for the whole 60-day period of the traffic collection; working on this step started in May and finished at the end of August 2018. In addition, the flow records were expanded to obtain specific bidirectional NetFlow data records, as the nfdump tool produces one direction by default, including date start/end time, IP source, IP destination, in packets (s2d packets), in bytes (s2d bytes), out packets (d2s packets), out bytes (d2s bytes), bps, pps, and bpp. The dumped file was divided each day and saved on a temporary database from day 1 until day 60. The data were divided into a daily format to establish daily samples for the users, thus organising the raw input data on a daily basis.

#### **4.4.2 IP source and MAC mapping**

A DHCP server leases an IP for a limited time and is highly likely to renew the leased IP after the lease time expires. Therefore, a given IP could be assigned to different computers at different times. Thus, to ensure no IP conflict occurred, the IP sources were mapped and a network adapter source MAC truth table was tagged in this step for each user on the investigated network using Python script. The IP source and MAC mapping process is not applied in any operational version of the proposed system; it is applied here simply to make sure there is no conflict between users. This process led to tagging the data for each user. The users were given numbers (from 1-23), which were saved on a database over the 60-day period, each day containing 23 users. This step was applied to match the

users on the investigated network with their daily traffic in order to be suitable for the proposed system’s novel methodology, and to make sure that the traffic was tagged to the exact user in an appropriate way; this process started at the end of August and finished at the end of October 2018. Therefore, the IP/MAC mapping mechanism was applied while ensuring that the profile was allocated to the correct host on the monitored network, as discussed in section 3.2.1. Since the MAC address of all hardware is unique, this makes MAC addresses more reliable than IPs, this approach ensuring that the IP sources did not change over time, according to the truth table. Table 4.1 shows a sample of a truth table for source MAC addresses along with the corresponding IP to keep track of the IP assignment analysed according to information derived from an ARP table, which was utilised for mapping IP source and MAC address control for users connected on the network. After the IP source addresses and source MAC were tagged, the IP/MAC address mapping was implemented. Figure 4.2 shows the number of flows for each user based on counting the flows using the unique IP source/MAC address. A total of 6,532,379 flows were collected from the lab. The output of this step is the start/end time, IP destination, in packets (s2d packets), in bytes (s2d bytes), out packets (d2s packets), out bytes (d2s bytes), bps, pps, and bpp. The output of the traffic data of this mechanism was saved to a database. The database contains the days from day 1 to day 60 and each user’s traffic data were tagged with the user ID (1-23) to be suitable as an input of the DNS system lookup mechanism.

**Table 4.1: MAC address and IP source mapping**

Timestamp	Source MAC	IP source
1526036411	b86b23eb1d7f	192.168.200.170
1526036411	b86b23000250	192.168.200.215
1526036411	b86b230e197f	192.168.200.129
1526058012	b86b23eb1d7f	192.168.200.170
1526058012	b86b23000250	192.168.200.215
1526058012	b86b230e197f	192.168.200.129

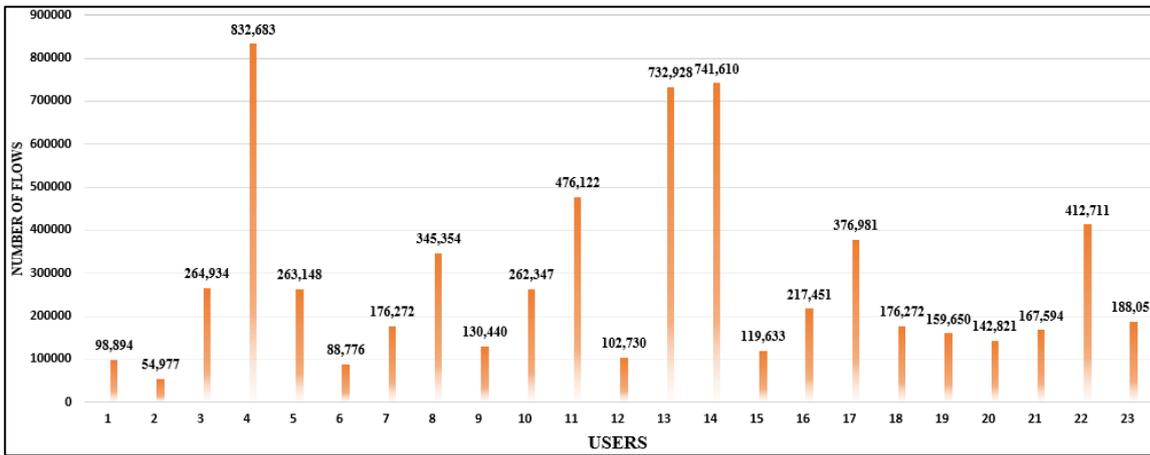


Figure 4.2: Number of flows for each user for the observed data

#### 4.4.3 Domain name system lookup

The DNS lookup utility was employed to identify the web applications used by the users (per flow). This was also used as a feature during the learning phase of the classification model and is explained later in section 5.2. The primary aim of using the DNS lookup utility in the system was to determine which flow belongs to which application and facilitates the automated application flow filtering process. After the data collection process, the associated domain names were resolved for each NetFlow record using Python script. This is in line with the use of DNS queries in several previous studies on tracking user behaviour and activity (McDowell, 2013) (Kirchler *et al.*, 2016), which indicates that relying on DNS queries is more accurate than relying on IP destination, as discussed in section 2.6.1. Each flow record contains the IP destination as the input for the mechanism start/end time, IP destination, in packets (s2d packets), in bytes (s2d bytes), out packets (d2s packets), out bytes (d2s bytes), bps, pps, and bpp. The corresponding domain name for each IP address was retrieved through Python script using the DNS lookup utility (Internet Systems Consortium, 2005) and was added as a new attribute to tag the application name (domain name) for each queried flow. This step was applied for each user during the 60 days, as mentioned earlier.

According to previous studies, IPs are subject to continuous change by the application owner for security reasons. Therefore, updating these addresses is essential when using this automated approach. The process took one-and-a-half months, from the beginning of November until the middle of December 2018. The process was utilised due to the use of NetFlow records, which do not retrieve DNS queries – they only retrieve the IP destination. Therefore, DNS queries were utilised versus IP destination for the predefined keywords to filter the web applications used in the next step. A converted domain name was added as a new attribute (DNS query) to the NetFlow records to be analysed in the next process of this study, as shown in Table 4.2. The output of this mechanism is as follows: the start/end time, IP destination, DNS queries, in packets (s2d packets), in bytes (s2d bytes), out packets (d2s packets), out bytes (d2s bytes), bps, pps and bpp, to form the input of the application flow filtering based on the DNS process.

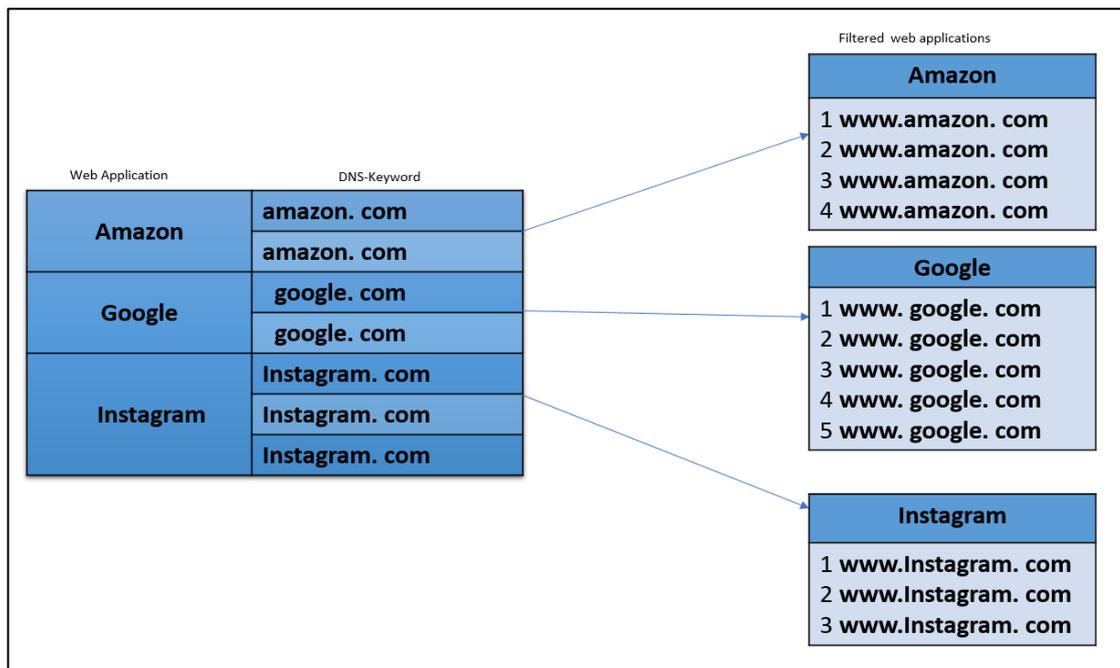
**Table 4.2: Features extracted after the domain name lookup process**

No.	Attribute	Description
1	date & time	Date and start /end time
2	IP dst	IP destination
3	in pkt	Source (client) to the destination (server) number of packets transmitted
4	DNS queries	The DNS query as 'bbc-vip016.cwwtf.bbc.co.uk.'
5	in byte	Source (client) to destination (server) bytes
6	out pkt	Destination (server) to source (client) number of packets transmitted
7	out byte	Destination (server) to source (client) bytes
9	bps	Source (client) to destination (server) bits per second
10	Pps	Source (client) to destination (server) packets per second
11	bpp	Source (client) to destination (server) bytes per packets

#### 4.4.3.1 *Application flow filtering based on domain name*

The flows were filtered and separated into groups of web applications after paring the corresponding flows to be adjusted (applications set) based on predefined keywords related to the 11 applications selected, which were the most connected web applications according to the statistical analysis applied to the network data of the users investigated in the research. The selected web applications were Amazon, Google, Instagram, Facebook, LinkedIn, Yahoo, MSN, Unknown, Stack Overflow, TeamViewer and IEEE.

The DNS query results were classified as an unknown category if the DNS lookup utility generator could not return any value for the given IP destination address. The input for this mechanism is the start/end time, IP destination, DNS queries, in packets (s2d packets), in bytes (s2d bytes), out packets (d2s packets), out bytes (d2s bytes), bps, pps and bpp. This mechanism was applied for all 23 users and 60 days of traffic data were saved on the database from the mechanism of the DNS system lookup during 20 days of processing time from the middle of December 2018. The application flow traffic was filtered and combined in data frames (similar to matrix object data) in terms of representing the usage and automating the process of dealing with the network traffic for each client's duration, as the data related to each user were separated in the previous steps. The output from this mechanism was saved on a database as 60 days, along with tagged traffic related to the 23 users. The filtered applications related to each user were also separated into groups with the following attributes: start/end time, IP destination, DNS queries, in packets (s2d packets), in bytes (s2d bytes), out packets (d2s packets), out bytes (d2s bytes), bps, pps and bpp to form the input for the flow session generator mechanism. The filtered data frame was used in the session generator step for the feature analysis, as illustrated in Figure 4.3.



**Figure 4.3: Filtered applications based on DNS**

#### 4.4.3.2 *Flow session generator*

The input of this mechanism is the database, which includes tagged traffic related to the investigated users and the filtered applications related to each user; as mentioned earlier, the applications were filtered and separated into groups. The input of this stage included the following attributes: start/end time, IP destination, DNS queries, in packets (s2d packets), in bytes (s2d bytes), out packets (d2s packets), out bytes (d2s bytes), bps, pps and bpp. The timing is the most important part of this process, as the concept of the flow session was implemented based on the start and end times of each session and separated by relying on the timing criteria to obtain the most discriminative features to identify and differentiate users. The data frames of the filtered applications were then analysed and divided into sessions using an assigned flow inter-arrival time threshold. A flow inter-arrival time was denoted by  $t$  (where  $t = \text{the start time of the second flow} - \text{the start time of the first flow}$ ) after converting the date and time to an epoch timestamp. The session concept applied led to extracting novel timing features to identify and differentiate users' 24-hour daily activity based on time bins represented by hour and quarter-hour time slots.

The session concept was calculated using a flow inter-arrival time threshold based on two conditions: the flows are part of the same session when  $t$  is less than the threshold (i.e., 10 seconds) and a new session starts when  $t$  is higher than the threshold (i.e., 10 seconds), as discussed in section 3.2.1. Furthermore, this procedure was applied in all filtered application data frames in order to divide each application into a set of sessions by generating features based on the session concept.

#### **4.5 Feature extraction process**

The feature extraction processes and their discriminative strength are paramount in maximising the accuracy of user identification. The statistical features and session timing-based resolution features were extracted to build datasets (hour and quarter-hour time bins) to form the input of the classifier. The content of the sample and exact features extracted presented in section 3.2.2 show that the hour session timing contained 101 features. In addition, each sample contained one application per user per day and the extracted statistical features (max, min, mean and median) were determined and included within the feature sets to add another dimension to the spacing of the features and was able to provide a user-dependent pattern, as shown in Table 3.1. The quarter-hour flow session timing features contained 172 features. Each sample contained one application per user per day to investigate the daily activity of the user from different levels to be compared with the hourly representation at the evaluation stage, as shown in Table 3.2. The statistical features extracted, which were calculated from the basic Netflow tuples, as discussed in section 3.2.3, were used to investigate their effect on the performance of the behaviour of the users and on the predicted results with the proposed flow session timing features (hour and quarter-hour). The proposed features were extracted for user identification and behaviour profiling to represent not only the application used by the user, but also the time slot on a daily basis. The full list of proposed feature datasets is shown in Table 3.3.

The dataset sample records were extracted from the original monitoring data collected from 23 users over 60 days as the network traffic. The total number of samples (records) in the two main datasets (hour and quarter-hour time bins) were 7,075 for all users, as one sample represents one application per user per day, which is a timing feature as well as an extracted statistical feature. The maximum number of samples was 736 for User 4 and the minimum number of samples was 25 for User 2; the mean number of samples for all users is 307, as shown in Figure 4.4.

The original data traffic was implemented and analysed over several steps through a traffic dump step, which mainly focused on dump traffic (nfcapd) files using the nfdump analysis tool. The dumped network traffic then proceeded with the IP source and MAC process to separate the traffic related to each user on a daily basis. The output of the IP source and MAC process proceeded with the DNS lookup process to identify the applications used by the user and utilised Python script. The output of the DNS lookup process was processed using Python script to filter the applications used by the user in order to be ready for the flow session generator step of extracting features and dataset records (samples). Sections 4.3 and 4.4 explain in more detail the time and exact inputs and outputs of each pre-processing step until reaching the dataset records (samples).

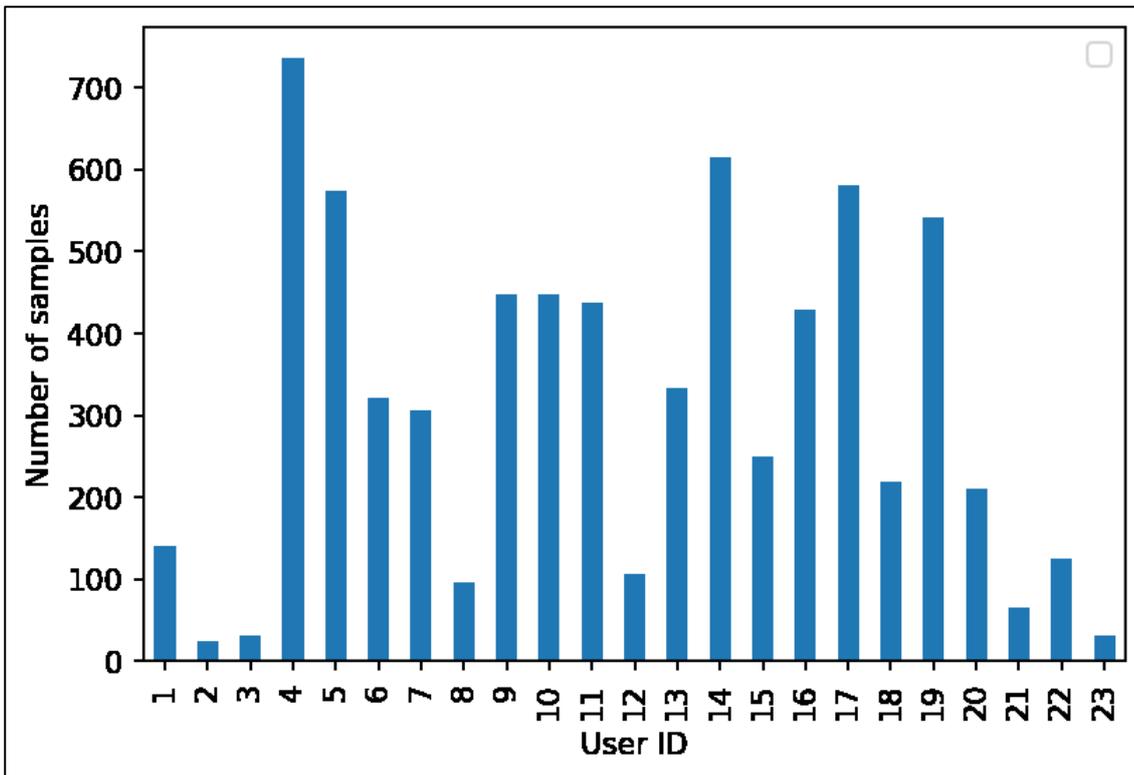


Figure 4.4: Number of samples for each user

#### 4.6 Gradient boosting classification

Once the network traffic metadata were pre-processed and the features extracted, only important features were selected on the basis of their contribution to the classification decision, which was decided by applying the random forest features importance and ranking algorithm. The generated features could be fed into a supervised ML algorithm, such as random forest, to compute each feature's importance to the decision being made. Random forest is one of the most commonly used algorithms for identifying feature importance to the selection process due to its low overfitting and easy interpretability. This is the core component of the proposed system, whereby an ML algorithm is used to classify users based on the selected features extracted from the network traffic according to the application-level flow session timing resolution to identify user behaviour and differentiate the daily usage activities between users. After developing the proposed method, a prototype experimental evaluation methodology could be used to measure the

accuracy and effectiveness of the model. The proposed datasets of extracted flow session timing features presented in section 3.2.2 were labelled with a user number after all the pre-processing steps were applied. Then, the list of features extracted (see Table 3.3) were randomly split into 70% for the training dataset and passed as inputs to the classifier to be trained; the other 30% represented the test dataset (unseen data) and remained to be classified. The training and testing sets were split randomly across samples (records) in a stratified fashion to ensure that the two sets contained approximately the same percentage of samples for each target user as the complete set. This method helps when using a training dataset to estimate unseen data and identify the quality of the proposed features. The classification method tests the quality of the features based on the gradient boosting classifier (learning algorithm) discussed in section 2.4.2. In other words, the classification method uses a gradient boosting learning algorithm, which has been used in previous studies and suits the specifications and characteristics of the classification dataset for evaluating the proposed features in this study. The gradient boosting classifier parameters (such as `n_ estimator` and `verbose`) were adjusted repeatedly until parameters were found that would work efficiently with the proposed features. The `verbose` parameter in the gradient boosting classifier progresses the performance of the classifier depending on the type of features fed to the classifier, as it is by default adjusted to 0 and, if adjusted to  $> 1$ , the progress performance increases for every tree. In addition, the `n_ estimator` parameter gave a better performance for the module by adjusting the number of boosting stages, which is by default adjusted to 100, and a larger number of the `n_ estimator` results in improved performance. Therefore, the gradient boosting classifier can work perfectly with a weak module by applying a sequence of iterations of classifications and repeated trees until reaching the best performance of the module. The best result predicted by the classifier was registered based on the percentage of overall accuracy, as well as a confusion matrix of the classifier results, and the best result was announced. Then, the

test dataset of the best result announced was implemented to calculate the probability ranks of the module to detect the true positive identification rate (TPIR). The gradient boosting supervised ML algorithm discussed earlier was applied to the proposed method for the extracted statistical features and flow session timing features. This was done to evaluate the system and obtain the most relevant results for identifying users and using the pattern of activities over 24-hour cycles to discriminate between users, as will be explained in the next chapter.

## **4.7 Discussion**

The proposed method is based on application-level flow session timing resolutions and uses NetFlow traffic information to extract relevant features to describe user behaviour. This approach should lead to a module that helps to identify users, not only by the daily 24-hour representation bins, but also the application used and the amount of traffic transmitted. The hour and quarter-hour timings are intended to enhance the analysis of the user traffic and offer the ability to keep user privacy and eliminate the limitations of the previous methods proposed in this field. In addition, relying on the processes presented in section 4.2 might enhance the way the network investigator deals with and monitors traffic.

Therefore, the observation and calculation of user behaviour activity among the different applications and the two different time slot features explained in section 3.2.2 offer an approach that uses DNS queries and user MAC address mapping to represent 24-hour daily user usage activity. The 24-hour daily user usage activity is represented by two different time bins (an approach that uses DNS queries to initialise applications and then identify user data in order to use session information to tag all traffic from that user), which could provide sufficiently discriminative information to identify users based on application flow-level analysis. The analysis relied on a truth table of MAC and IP

addresses to ensure host consistency, as the DHCP will change IP addresses among users during that time. The proposed method was evaluated by experiments using datasets based on the proposed timing features to build a dataset containing the full list discussed in section 3.2.2 as an input of gradient boosting supervised ML to investigate the effect of the proposed features on user identification and behaviour profiling. The data were collected from the network switch, which indicates that the traffic data were collected while the user was browsing his/her machine in a typical manner and without any interaction from the researcher. The data collected were processed and analysed, starting from the data collections until the classification step, to evaluate the system and investigate the variance and similarities between users in the module. The way in which the traffic was collected and the several processes implemented for the user traffic give the module a strong method for dealing with user traffic. This should enhance the ability of the investigator in an organisation to deal with user traffic, identify users and make informed decisions about security or how to improve policy.

## **4.8 Conclusion**

This chapter presented the methodology and data collection methods employed in this research as well as the general block diagram of the proposed system for user behaviour profiling using application-level flow session timing features from generic network traffic. The proposed novel set of features is based on application-level flow sessions (inter-arrival time between flows) and DNS query filtering in addition to IP/MAC address mapping. This set of features was fed to a supervised ML algorithm based on the NetFlow network traffic captured from CSCAN at the University of Plymouth to investigate the proposed approach for 60 days with 23 users. The processes used for collecting the data and analysing the user traffic in an automated way when the user is engaged in typical browsing on his/her machine were explained in a series of steps. The different steps presented in section 4.2 were utilised to obtain datasets of the proposed features and label

them with the user number to represent the traffic launched by the user, as explained in the sections above, ready for the evaluation and analysis steps outlined in the next chapter.

# **Chapter Five**

## **Evaluation and analysis**

## 5 Evaluation and analysis

### 5.1 Introduction

This chapter presents the evaluation and analysis of the proposed system based on the timing features used and presented in section 3.2.2. The system was evaluated using the gradient boosting algorithm, which is explained in section 2.4.2, to classify traffic based on the features extracted and the pre-processed dataset, as explained in the previous chapter, in that two different datasets were extracted based on flow session and timing resolution criteria. The input dataset was applied to the gradient boosting algorithm due to its ability to produce a series of trees to correct any errors that occurred with the previous tree. The Z-score was applied to the dataset to normalise the numeric data, excluding the binary bin features to enhance the classifier in the end classification model (Yang, 2010). The data were split randomly into two sets: 70% of the data were used to train the gradient boosting classifier and 30% were used to test all the users' data. The classifier performance was evaluated using different metrics derived from the following four parameters: true positive (TP), false positive (FP), true negative (TN) and false negative (FN). The evaluation parameters (accuracy, precision, recall and F1 score) were calculated based on error rates, which are represented in the confusion matrix according to the following equations:

- **Accuracy:** predicts the overall accuracy of the model:

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

- **Precision:** gives the fraction of the classifier prediction that is true:

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

- **Recall:** the percentage of true results out of all the results estimated by the classifier:

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

- **F1 score:** a metric conducted by calculating the precision and recall. This is more useful than accuracy in the case of uneven multiclass distribution and if the FP and FN are very different (Vinupaul *et al.*, 2017):

$$F1\ score = \frac{Recall*Precision}{Recall+Precision} \quad (4)$$

In addition, the TPIR was applied to obtain the probability ranks; if each user sample output by the system for the identities top probability match  $\alpha$  where  $\alpha$  represents the rank of accuracy. Probability ranks were applied to allow the system to attempt to determine the identity of an individual. The probability ranks were implemented based on testing each sample in the dataset and comparing the true user ID with five different probability ranks. If a true user exists in rank 1, TPIR1 takes the value 1; if the user ID exists in rank 1 and rank 2, then TPIR2 takes the value 1 and so on, based on the following equation:

$$ranks = if\ (true\ UID)\ in\ rank1\ TPIR1 = 1\ if\ not\ TPIR1 = 0$$

## 5.2 User identification and profiling using application-level session hourly timing features

This section presents the results of the first experiment, which was based on the flow sessions and hourly session timing resolution features explained in section 3.2.2. The dataset was implemented with a 10 sec flow inter-arrival time threshold based on the distribution analysis explained in section 3.2.2. The flow inter-arrival time distribution analysis was applied to reach the most useful dataset with the highest performance based on the proposed method, as mentioned through the experiment when the highest

distribution conducted occurred with 10 sec. In this experiment, different datasets were implemented based on the 10 sec session threshold described in section 3.2.2 and a range of periods to investigate the most relevant criteria to build a system with the ability to identify and profile users. The dataset was divided into three sets (set 1, set 2 and set 3) due to the variance in the volume of traffic during the data collection to investigate the variability of user behaviour over time in the context of the dynamic and changing nature of network applications. Set 1 represents the first month of collected user traffic; set 2 represents the second month of collected user traffic; and set 3 represents the whole 60 days of collected data. The volume of traffic between these different sets affected the performance of the classifier, as the accuracy differed between the first month, the second month, and the whole two months. The accuracy kept increasing when the periods of the datasets changed, as set 2 (which represents the second 30 days) scored the highest accuracy compared with set 1 (the first month of data) and set 3 (the whole 60 days); this is explained in detail in the next section. The variance between the sets provides discriminative information to differentiate users' behaviour based on the proposed hourly timing features to represent the 24-hour activity of the users in the investigated system.

### **5.2.1 User identification rates: experimental results**

The results shown in Table 5.1 reveal that the flow-based network traffic analysis and statistical features can produce a notable result in terms of user identification and behaviour profiling. In comparison with previous studies (McDowell, 2013; Vinupaul *et al.*, 2017), in which the accuracy achieved was 66% using a flow network analysis approach, this study achieved a level of up to 86% accuracy when identifying users as whole populations based on the proposed hourly flow session timing features datasets. Therefore, different aspects affected the accuracy of this study when compared with other studies, such as the volume of traffic and the environment in which the flow network traffic was collected. For example, one study (McDowell, 2013) collected traffic from the

Computer Science building at a Naval Postgraduate School for 6 days, which is similar to the test environment in this thesis. Users were identified using DNS queries and the number of connections was similar to the approaches utilised in this thesis. In contrast, this thesis utilised DNS queries and the number of connections as well as timing features, which were extracted, selected and validated by collecting real network traffic during a 60-day period, which is ten times longer than the period examined by McDowell (2013). Therefore, the variance in the number of days between this thesis and the compared study had a positive effect on the results, which are demonstrated as being 20% higher than the aforementioned study. Other researchers (Vinupaul *et al.*, 2017) collected traffic in a university network from volunteer users using information-gathering software for 7 days. However, the volunteers may consciously or unconsciously have altered their behaviour and installing information-gathering software on each user's machine might not have been an accurate way of collecting network traffic. In contrast, this thesis sought to identify and profile users without using volunteers or information-gathering software. Although the studies referred to above and compared with this thesis used very similar test environments (a university laboratory), the difference lies in the number of days: 60 days in the case of this thesis. The number of days for which traffic was collected had a positive effect on the accuracy of the proposed system compared with the previous studies.

The three different sets presented in the previous section were applied to the classifier separately by training the classifier based on 70% of set 1. Consequently, 30% of set 1 was tested as unseen data to investigate the quality of the proposed features, as well as the effect of the application on the first month of user traffic investigated. Set 2, which represents the second month of the data, was applied separately to the classifier in the same manner as set 1 to investigate the effect of both sets. Set 3 was applied by combining the traffic that related to the whole of the 60 days to compare the effect of two full months of traffic on the system. Table 5.1 shows the accuracy of the sets. The first row of the

table shows the accuracy ranks of set 1, which represents the first 30 days of the investigated data. The second row shows the accuracy of set 2, which represents the second 30 days of the data; this exceeded the accuracy of set 1 and set 3, which represent the first 30 days and all 60 days of the data. Therefore, the data in set 1 and set 3 were less affected by the traffic generated by the experimental lab users (this was due to a holiday), whereas the set 2 traffic generated by users was normal, which affected the volume of the interactions of the users included in the sets. The highest accuracy for set 2 improved the proposed measurement features that were affected by the periods of data collection and user access limitations, which were observed in the volume of traffic data between all sets. For set 2, which represents the second 30 days, it can be seen from the collected data scores that the highest TPIR (74%) was in rank 1; this value gradually increased to 77% in rank 2, 80% in rank 3, 84% in rank 4 and continued to improve up to 86% in rank 5. The results for set 2 demonstrate that it is possible to use hourly flow session timing features for user identification and profiling as the highest unit of time, as discussed in section 3.2.2. The average accuracy between ranks, starting from rank 1 until rank 5, increased, which indicates that the discriminative information regarding the proposed timing features caused the probability of the ranks to increase when the testing phase was analysed.

**Table 5.1: Hourly timing features - users' traffic identification rate results**

Sets	No. days	Rank1	Rank2	Rank3	Rank4	Rank5
Set 1	1st/30	70%	73%	73%	74%	77%
Set 2	2nd/30	74%	77%	80%	84%	86%
Set 3	60	72%	75%	78%	78%	81%

Furthermore, the 60-day duration of this experiment, represented by the set 3 data, scored the second-highest TPIR accuracy (72%) in rank 1 and this increased slightly to 75% in rank 2; this continued to increase until reaching 78% in rank 3, 78% in rank 4 and the highest accuracy reached was 81% in rank 5. The first 30 days, represented by set 1,

scored the lowest TPIR accuracy in the system due to several issues concerning effectiveness, as explained at the beginning of this section. Set 1 scored the lowest TPIR accuracy between set 2 and set 3 with 70% in rank 1, compared with 72% and 74% for the other sets in the system. In addition, the accuracy of set 1 increased to 73% in rank 2, 73% in rank 3, 74% in rank 4, and continued to rise to 77% in rank 5. Therefore, the TPIR accuracy for set 1 increased only slightly and the accuracy for rank 2 and rank 3 did not increase compared with set 2 and set 3, which is where its TPIR continued to increase, starting from rank 1 until reaching rank 5. As a result, the experiment shows that there are several components that affect the system and its ability to identify and differentiate between users. These components varied between the time of collecting the data and the environment in which the data were collected. These components can clearly be seen in the variance in TPIR accuracy between the three different sets implemented in this experiment.

In addition, rank 1 for TPIR accuracy between the three sets increased slightly, starting from 70% in set 1, 72% in set 3 and 74% in set 2, which indicates that set 2 has discriminative data to identify users on the system. Rank 2 TPIR increased by 2% from set 1 to set 3 and continued to increase by 3% in set 2, which makes a slight difference in this rank between the three sets. Rank 3 showed a big difference between the three sets, as the TPIR accuracy increased by 5% from set 1 to set 3 and 2% from set 3 to set 2, which is the highest of the three sets ranked. The results for rank 4 rose by 10% from set 1 to set 3, continued to increase by 4% from set 1 to set 3, and increased by 6% from set 3 to set 2. Rank 5 TPIR scored the highest accuracy of all the sets, as it reached 86% based on the set 2 data, starting from 77% for set 1, which indicates that set 2 contains discriminative information to identify users on the system. The variance in the accuracy of the ranks between sets indicates that the system has the ability to differentiate between all the populations on the system, even if the amount of traffic affects the results between

the ranks, with an average for all sets ranging between 3% and 5%. The results of this experiment demonstrate that using flow session timing features derived from the application level can help to create an accurate user identification system. Subsequently, the system was able to identify the whole population based on set 2 (the second month of investigated data), with an 86% level of accuracy.

### **5.2.2 Individual user TPIR ranks**

With regard to identifying individual users, Table 5.2 shows the TPIR results for each user's probability ranks to show the ability of the system to identify users based on the proposed method and hourly timing features extracted and implemented in this experiment. The performance of the classification model varied between users on the system in that the TPIR accuracy ranks differed. User 10 showed the highest accuracy of all the users on the system, with 97% in rank 1, 98% in rank 2, continuing to increase until reaching 99% in rank 3 and remaining stable at 99% in rank 5. The second-highest accuracy on the module was scored by User 6, with 91% in rank 1, 93% in rank 2, and continuing to increase until reaching 99% in rank 5. Therefore, the high level of accuracy of the top two users in the module indicates that the proposed method and extracted features have sufficiently discriminative traffic information to identify and differentiate between users and the applications used for the system to build a significant profile of the user. The third-highest TPIR accuracy recorded was for User 12, with 89% in rank 1, 90% in rank 2, and 91% in rank 3, which continued to improve until reaching 94% in rank 5. Some of the users scored medium TPIR accuracy and others scored low TPIR accuracy on the system. Some users achieved a low TPIR in ranks 1 and 2 but good improvement up to almost 70% correct classification of their traffic in rank 5. For instance, User 23 achieved a 48% TPIR in rank 1; this proportion continued to increase in each rank to attain almost 69% of the traffic being accurately classified.

**Table 5.2: Hourly timing features – users’ TPIR ranks (features set 2)**

User	Rank1	Rank2	Rank3	Rank4	Rank5
1	67%	68%	76%	82%	86%
2	85%	87%	89%	90%	92%
3	90%	92%	92%	95%	97%
4	70%	74%	79%	80%	83%
5	58%	73%	74%	76%	80%
6	91%	93%	96%	99%	100%
7	75%	77%	80%	81%	83%
8	72%	79%	82%	85%	87%
9	75%	78%	80%	83%	85%
10	97%	98%	99%	99%	99%
11	87%	88%	92%	94%	95%
12	89%	90%	91%	93%	94%
13	57%	64%	70%	74%	75%
14	71%	76%	77%	78%	78%
15	71%	78%	79%	84%	86%
16	68%	68%	76%	77%	83%
17	78%	80%	83%	85%	87%
18	88%	90%	91%	92%	93%
19	65%	67%	72%	74%	76%
20	71%	73%	76%	83%	86%
21	60%	65%	67%	77%	80%
22	58%	60%	65%	79%	83%
23	48%	50%	50%	65%	69%

It is clear that one third of users achieved a TPIR of 50% and above in rank 1, but this proportion rose dramatically to three quarters of the participants achieving a 50% TPIR. Therefore, the majority of the users can be correctly identified in rank 5 by more than 80%, as can be seen in Table 5.2. However, the system could not find discriminative information from the traffic of one of the 23 users, and he/she achieved under 50% TPIR in rank 1. Although some users achieved a low level of correct classification in rank 1 and rank 2, they achieved a better level of accuracy in the remaining ranks, such as users 5 and 23. There is a promising TPIR accuracy score for the users on the system using the proposed method and the hourly features extracted at the level of the individual user on the investigated network, as the system could identify and differentiate between users based on their discriminative information.

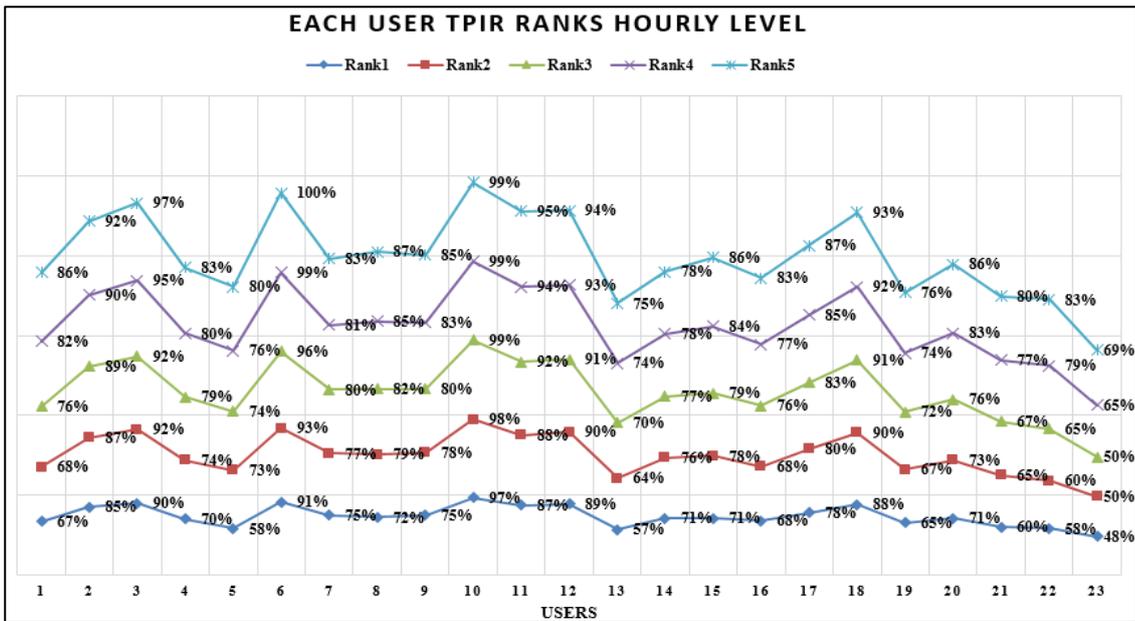


Figure 5.1: Hourly timing features – individual users’ TPIR

Figure 5.1 shows the TPIR frequency between the users on the system. Users 3, 6 and 10 scored the highest frequency among the 23 users on the system based on all ranks. Other users scored a medium frequency; as shown in the figure above, the accuracy of users 5, 8, 15 and 21 continued to increase based on rank 1 until reaching rank 5. The users on the system who scored a low frequency based on their TPIR rank might have done so because they did not have discriminative traffic information through which the system could identify them; for instance, users 13 and 23 showed low accuracy for all ranks. In contrast, the user identification system had the ability to identify an individual user with a high level of accuracy based on the rank 1 results. The highest identification rate scored was for User 10, which reached a 97% level of accuracy with a 49% TPIR that was higher than the lowest user score in rank 1. User 10 achieved the highest score accuracy from among the users, which indicates that the user has more interaction usage based on the time bin features compared with the other users on the system. In rank 1, User 1 scored 67%, which gives an indication of the variance between users on the system; for example, User 2 scored 85%, which is higher than User 1 by 18%, although this shows less difference when compared with the highest user score on the system based on rank 1 and

the lowest user on the system. User 7 scored an identification rate of 75% based on rank 1 with a difference of 3% between his/her score and User 8's score, which indicates that their discriminative information is close in variance and similarity due to the system identifying them with an almost similar score. In addition, it is clearly shown that the identification rate for User 9 is similar to that of User 8 with a score of 75%, which indicates that the system identified them with a similar identification rate based on their similar usage. User 21's score and identification rate was 60% compared with User 22, who scored 58%, which indicates a difference between them of 2%, which gives an indication of their similarities in usage-based application-level hourly timing features.

In comparison, the system identified users in rank 2 with a high accuracy. For example, User 10 scored a frequency of 98%, which was 48% higher than the lowest user on the system based on rank 2 (User 23). User 7 scored an accuracy of 77% in rank 2, which increased by 2% from rank 1, compared with User 8, who scored 79% in rank 2, which is an increase of 7%. The improvement in the accuracy between users indicates that User 8 has a significant increase compared with his/her accuracy in rank 1, which is better than User 7 based on their variance in usage. As a result, the improvement in accuracy between users in each rank indicates that the proposed system has the ability to identify users' 24-hour activity based on hourly timing features. This provides an accurate system with an accuracy rate of 100%, such as User 6, based on the TPIR.

### **5.2.3 Confusion matrix**

The set 2 dataset was implemented and analysed from different perspectives based on individual user implementation, using a confusion matrix with precision, recall and F1 score based on the equation explained in section 5.1. Table 5.3 shows a confusion matrix for all users to demonstrate the correct and incorrect prediction of each class based on the test data for the set 2 features set. The performance of the classification model is fairly

high among all the classes, ranging from 48-100%. The labels indicate the users' ID (from User 1 to User 23), as illustrated in Table 5.3, for the predicted and true labels.

The highest score of the actual class was recorded by User 10 (97% of the TP classified samples), with 3% of the FN recorded for users 4 and 12, which shows the ability of the module to identify users with a high score. Furthermore, User 6 recorded the second-highest accuracy score among all the users (91% of the TP classified samples); however, there was an FN figure of 9% for User 23. User 3 recorded the third-highest score for the model with 90% TP and 14% of the FN attributed to users 2 and 20. User 23 recorded the lowest accuracy with 48% TP and 52% FN attributed to users 2, 4, 7, 13 and 18 on the model because these users had the lowest number of traffic samples. The reason for the small number of traffic samples for User 23 was that this user was not active on each of the 60 days, which affected his/her TP rate compared with the others.

**Table 5.3: Hourly timing features confusion matrix (features set 2)**

Uid	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	67	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0
2	0	85	0	0	5	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	5	0	0
3	0	6	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
4	0	0	0	70	0	0	0	0	0	0	20	0	10	0	0	0	0	0	0	0	0	0	0
5	0	0	0	5	58	0	0	0	6	0	0	0	0	0	5	21	0	0	0	5	0	0	0
6	0	0	0	0	0	91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9
7	0	0	0	0	0	0	75	0	8	0	0	0	0	0	0	0	17	0	0	0	0	0	0
8	5	0	0	0	5	0	0	72	0	0	0	3	0	0	0	0	10	0	0	5	0	0	0
9	0	0	0	0	0	0	7	0	75	0	0	0	0	0	0	7	0	0	7	0	4	0	0
10	0	0	0	2	0	0	0	0	0	97	0	1	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	3	0	0	0	0	0	0	87	0	0	0	0	7	0	0	0	3	0	0	0
12	0	0	0	11	0	0	0	0	0	0	0	89	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	7	0	0	7	0	0	57	0	0	8	0	0	0	21	0	0	0
14	0	0	0	14	0	0	8	0	0	0	7	0	0	71	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	15	0	0	71	0	0	14	0	0	0	0	0
16	0	0	0	11	0	0	6	0	6	0	0	0	6	0	0	68	0	0	3	0	0	0	0
17	0	0	0	4	4	0	4	0	0	0	6	0	0	4	0	0	78	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	88	2	0	0	0	0	0
19	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	18	0	65	0	0	8	0
20	0	0	0	8	0	0	0	0	8	0	0	0	0	0	7	0	0	0	0	71	0	0	6
21	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	10	0	0	60	0	0	0
22	0	0	15	0	0	0	0	10	0	0	0	2	0	0	0	0	0	15	0	0	0	58	0
23	0	12	0	15	0	0	10	0	0	0	0	0	10	0	0	0	0	5	0	0	0	0	48

In addition to the TPIR accuracy ranks implemented for the individual user to encourage the system to identify users, referred to in section 5.2.2, different measurements were applied to the system to analyse individual users based on information derived from the confusion matrix. Therefore, set 2 data were implemented with different measurements (recall, precision and F1 score), which were extracted from the confusion matrix TP, FP and FN based on the equation explained in section 5.1. The measurement used is commonly applied to several classification problems. Figure 5.2 shows the results of the implementation of set 2 data to assess the difference between users from a different perspective. User 1 scored 93% precision and 67% recall, which indicates the FN is higher than the FP and TP rates. User 2 scored 85% recall and 74% precision, which indicates that the FP is higher than the FN and TP rates. Therefore, when the precision is higher than the recall, it means that the FN is higher than the FP and TP, which gives an indication that the user has been identified with a reasonable score. As a result, the F1

score measurement gives a balance between recall and precision, as it is calculated based on these two measurement components.

For comparison, the analysis of the recall, precision, F1 score and TPIR rankings for individual users is discussed in section 5.2.1. It is clearly shown in Figure 5.2 that User 10, who scored the highest TPIR accuracy among all the users on the system, also scored the highest for recall (97%), precision (85%) and F1 score (90%). As the recall is higher than the precision score, in the case of this system, it means that the TP and FP are higher than the FN, which indicates that the system identified User 10 with a high level of accuracy, as demonstrated by the results of the TPIR and the confusion matrix analysis. User 23, as shown in Figure 5.2, scored the lowest of all the users on the system, with 48% for recall, 76% for precision and 58% for the F1 score. As precision scored a higher frequency than recall, the FN is higher than the FP and TP, which indicates that the user sample was assigned to other users on the system. The TPIR results discussed in 5.2.2 show that the system identified User 23 with the lowest accuracy among the users on the system, which is also demonstrated by the confusion matrix and the recall, precision and F1 score analysis.

Based on the confusion matrix and recall results, the precision and F1 score analysis implemented on the system indicates that when a user scored higher for precision than for recall, the FN for the user is higher than the FP and TP, which means that most of the user samples went to another user on the system. On the other hand, if the recall is higher than the precision, this means the TP and FP are higher than the FN, which indicates that the system has identified the user with a high level of accuracy from among the other users due to most of the user sample being correctly assigned to a particular user on the system. The results explained for the highest and lowest users on the system were applied to the

other users by taking the same hypothesis to demonstrate the quality of their score results on the system.

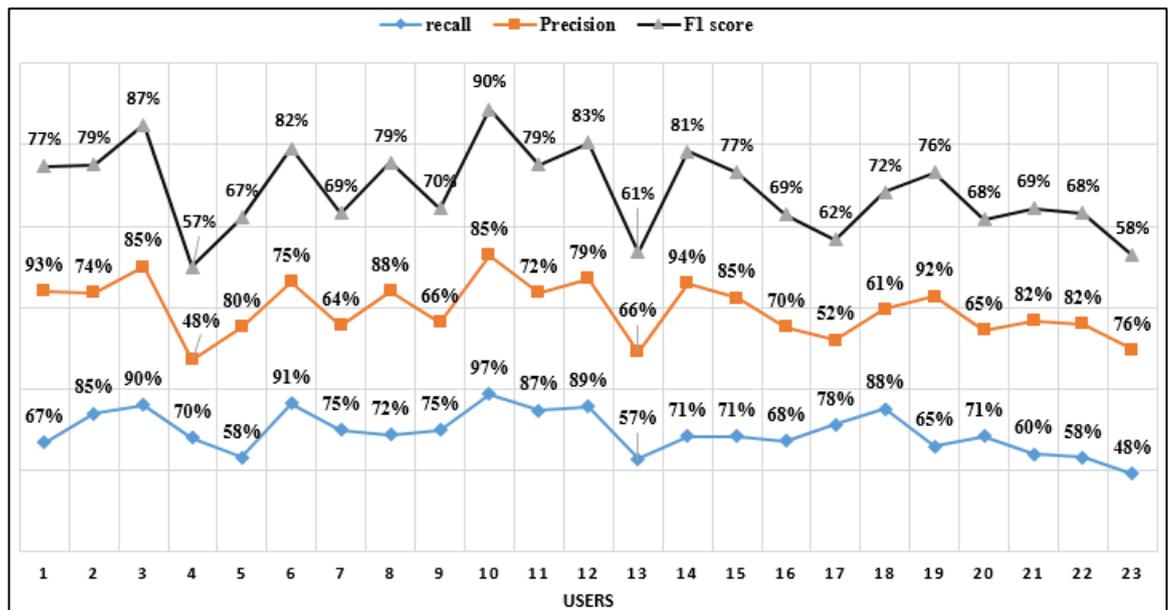


Figure 5.2: Hourly timing features for individual users (recall, precision and F1 score)

#### 5.2.4 Feature importance

Feature importance analysis was implemented in this experiment to assign weights to the features and identify those with the greatest effect on the performance of the system. The random forest technique was employed on the set 2 dataset, which produced the best performance and the highest accuracy in this experiment based on the hourly timing features presented in Table 3.1. Table 5.4 shows the weights of the top 40 features on the module out of 101 features extracted and implemented in this experiment. The top 40 features are represented in Figure 5.3. The first four features (app\_encoded, end\_hour, start\_hour, and number\_of\_connections) are the most influential among all the extracted statistical and proposed hourly session timing resolution features for the whole dataset. This finding is in line with what was expected, in that different users have different usage patterns in terms of applications and the time of use.

The proposed timing features show the ability to better discriminate between users in comparison with other parameters, which enhances the classifier's capability. Table 5.4 shows the ranking weights for the session timing features and the basic statistical features for most of the attributes that were used by the gradient boosting classifier. The 'app\_encoded' feature scored the highest rank weight (0.05027807), which was stored in the dataset as 0-10, and the encoded integer enhanced the module's ability to identify users on the system by strongly enhancing the classifier's ability to differentiate between users. The 'app\_encoded' feature was specifically extracted based on the DNS lookup process discussed in section 4.4.3, which indicates the efficiency of this process. The findings based on the ranking weights analysis suggest that the specific applications used by each user provide a high indication/bias to identify the user.

In addition, the 'start\_hour' and 'end\_hour' features scored the second- and third-highest of all the features, which highlights the importance of hourly timing resolution features in identifying users from their investigated traffic samples. The 'end\_hour' feature scored the second-highest ranking weight on the system (0.02890537), which was stored as the end timing of the application usage (0-23) during the 24-hour daily user activity, and was a major enhancement for the classifier to identify users after the 'app\_encoded' feature. The 'start\_hour' feature scored third for rank weight at 0.02479219, which was stored as 0-23 to represent the end time of the application used during the 24-hour daily usage activity.

Therefore, the two features above were ranked according to a high weight, which gives an indication that the representation of the timing slots for each user for specific applications gives the correct differentiation, and improves the ability of the classifier to identify the user after the 'app\_encoded' feature is implemented. The 'no\_of\_conn' feature, which represents the amount of data exchanged on the dataset, scored the fourth-ranked

weight of 0.02437863. The high-ranking weight of the number of connection feature indicates that the representation of the data exchange shows the ability of the classifier to differentiate correctly and identify users on the system. In contrast, the binary encoding of the ‘start\_hour’ and ‘end\_hour’ features discussed in section 3.2.2 enhanced the classifier’s ability to identify users based on the ranking weights, which suggests that the decimal feature hours (7, 10, 6, 8, 9) presented in Table 5.4 are among the top 40 features, as the time slots are stored on the dataset and take 1 if the time slot is used and 0 if the time slot is not used, as shown in Table 3.1. The weights of the top 40 features on the module indicate that the proposed method and features enhanced the classifier’s ability to differentiate between users with a good score, as explained in section 5.2.1. Furthermore, the timing features enhanced the system’s discriminative information regarding the user’s traffic information, which encouraged the classifier to identify users, with promising results.

**Table 5.4: Hourly timing features importance weights**

<b>Feature</b>	<b>Ranking weight</b>	<b>Feature</b>	<b>Ranking weight</b>
app_encoded	0.05027	5	0.01277
end_hour	0.02890	median_D2S bpp	0.01276
start_hour	0.02479	sum_out pkt	0.01274
no_of_conn	0.02437	min_out byte	0.01252
max_in pkt	0.02102	min_D2S bpp	0.01237
mean_in pkt	0.01840	8	0.01208
sum_Recived_to_transsmited	0.01742	mean_out byte	0.01165
sum_D2S bpp	0.01741	sum_D2S pps	0.01160
max_out pkt	0.01736	median_bpp	0.01114
sum_bpp	0.01699	9	0.01106
min_Recived_transsmited_data	0.01696	sum_pps	0.01100
mean_out pkt	0.01683	min_bpp	0.01067
7	0.01657	median_D2S pps	0.01066
min_Recived_to_transsmited	0.01551	min_in pkt	0.01037
10	0.01524	median_in byte	0.01027
sum_in pkt	0.01339	min_D2S pps	0.01024
mean_D2S bpp	0.01294	min_Transmitted_DR	0.01016
mean_in byte	0.01294	max_out byte	0.01015
6	0.01291	min_in byte	0.01014
median_Recived_transsmited_data	0.01289	median_in pkt	0.01009

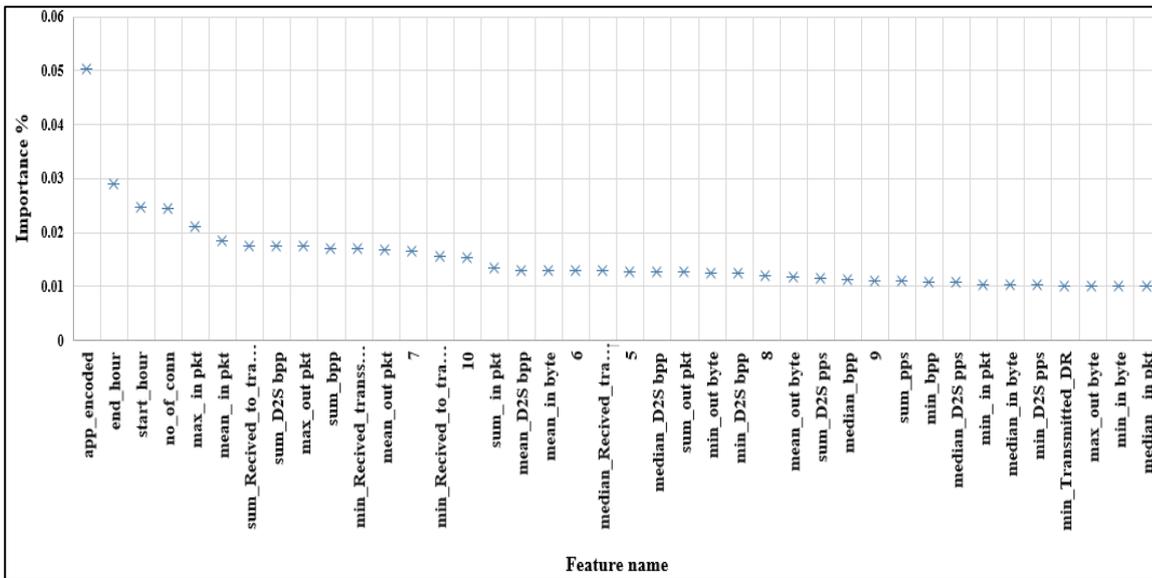


Figure 5.3: Hourly timing features scores

### 5.2.5 Descriptive analysis of mean and standard deviation variance

It is widely known that descriptive analysis is capable of explaining the main features of a dataset. Therefore, the statistical descriptive analysis approaches implemented for the proposed and extracted features are based on the most important features employed in the previous section. Descriptive analysis of the features is implemented to present the individual users' patterns as presented by the dataset that was extracted and processed, as shown in the previous chapter. Descriptive analysis is employed to show that the proposed features contain sufficiently discriminative traffic information to differentiate and identify users from different perspectives. The mean and standard deviation (SD) analysis was implemented to identify the variance and describe the usage amongst the users' features. Table 5.5 shows the mean and SD descriptive analysis frequencies by taking the top five statistically important features that were presented by the feature importance analysis explained in section 5.2.4. The descriptive analysis is based on the mean and SD applied to show the variance and difference between individuals based on the discriminative information provided by the novel timing features extracted from the user identification system.

**Table 5.5: Hourly timing features - top five mean and SD variances**

Users		app_encoded	end_hour	start_hour	no_of_conn	max_inpkt
1	Mean	2.81	12.86	12.48	4.76	80.61
	SD.	2.57	6.22	6.16	4.82	222.12
2	Mean	2.58	14.54	14.33	5.11	29.45
	SD	2.45	5.36	5.51	4.54	59.34
3	Mean	2.11	13.89	13.64	2.26	40.48
	SD	2.33	4.58	4.53	1.09	70.65
4	Mean	4.08	13.50	10.81	13.85	990.27
	SD	3.24	4.75	4.74	25.29	4250.59
5	Mean	4.32	14.44	10.11	8.71	960.56
	SD	3.36	5.47	5.60	14.92	3778.14
6	Mean	2.63	13.00	12.42	3.19	304.23
	SD	2.81	3.72	3.57	5.35	3447.68
7	Mean	4.16	14.12	13.08	9.14	277.12
	SD	3.51	2.04	1.93	11.32	1529.23
8	Mean	5.03	9.75	7.82	140.44	29.39
	SD	3.98	5.90	4.86	262.28	46.64
9	Mean	3.13	14.93	13.80	4.88	38.41
	SD	3.18	3.41	3.02	3.97	70.62
10	Mean	4.35	13.20	11.25	12.66	76.97
	SD	3.33	3.21	2.81	12.60	371.39
11	Mean	4.38	14.58	12.29	11.70	345.43
	SD	3.25	4.70	4.31	11.61	1412.10
12	Mean	2.50	13.70	11.95	13.28	178.89
	SD	3.14	6.21	6.13	61.23	263.52
13	Mean	2.48	16.44	7.91	12.01	393.72
	SD	3.14	5.65	5.88	12.30	2062.02
14	Mean	3.70	13.06	10.99	8.06	229.99
	SD	3.06	3.56	3.13	7.14	567.14
15	Mean	3.22	15.64	14.49	8.61	196.96
	SD	2.71	4.59	4.72	11.60	819.88
16	Mean	4.64	13.46	11.78	6.26	137.80
	SD	3.41	3.89	3.08	5.50	671.72
17	Mean	3.56	16.07	13.55	11.88	101.23
	SD	3.07	4.69	5.43	25.81	434.90
18	Mean	3.56	12.29	11.34	9.40	52.45
	SD	2.99	1.71	1.59	7.61	85.00
19	Mean	4.22	16.12	14.60	16.99	117.73
	SD	3.20	3.01	2.77	18.83	402.48
20	Mean	0.84	17.34	6.61	10.31	48.81
	SD	1.28	4.69	5.66	12.16	256.62
21	Mean	1.27	13.37	11.44	2.46	71.17
	SD	1.68	3.88	4.62	2.81	103.27
22	Mean	3.58	15.32	13.89	11.30	28.51
	SD	3.00	2.57	2.66	15.11	28.91
23	Mean	6.25	19.19	15.69	180.27	2018.69
	SD	7.62	8.32	7.06	400.13	10296.41

Table 5.5 shows that users 1, 2, 4 and 15 shared quite similar frequency in terms of ‘app\_encoded’ mean and SD variance. However, there is a considerable difference between these users and the other users on the system based on this type of feature. For instance, User 1 and User 16’s mean and SD variance have different frequencies, which gives a degree of discrimination for a classifier to identify them. The variance between users for ‘app\_encoded’ feature indicates that the users have discriminative traffic information and used different applications at a different time, which gives the system and the features extracted a strong motivation to differentiate between users. Moreover, observing the frequencies between users on this feature indicates that User 23 scored the highest mean and SD frequency among all the users on the system. This might affect the TPIR for this user among the other users based on the TPIR ranks for User 23, which were the lowest on the predicted module. User 10 has the lowest mean and SD frequency on the ‘app\_encoded’ feature, which might affect the TPIR for this user on the system.

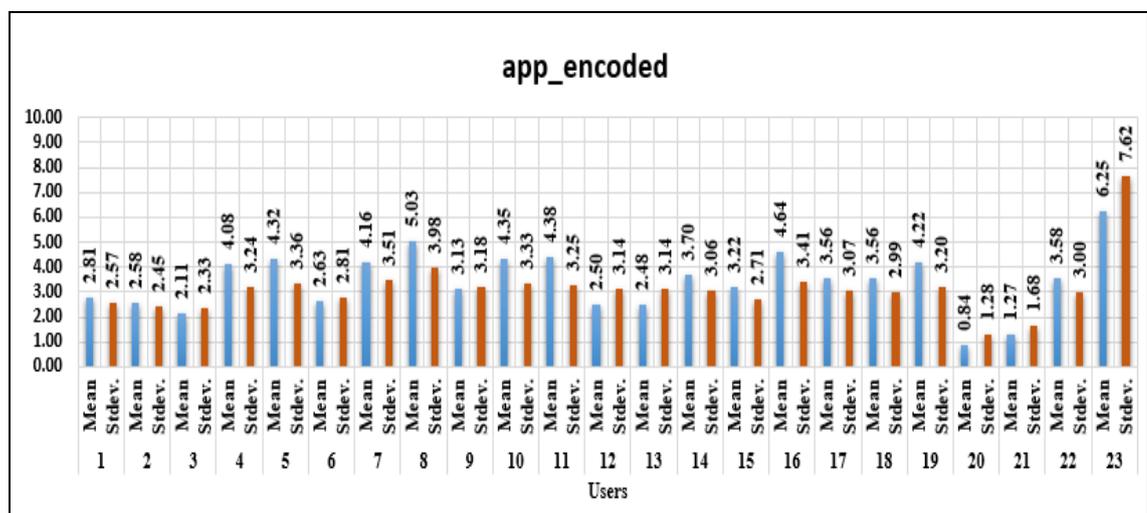
However, analysing other features reveals that users 3, 4, 6 and 10 have a similar mean and SD for the ‘end\_hour’ feature, as well as users 16 and 21 having an almost similar frequency for the same feature. On the other hand, User 23 has a unique frequency compared with all the other users on the system, which might affect the performance of this user compared with the others.

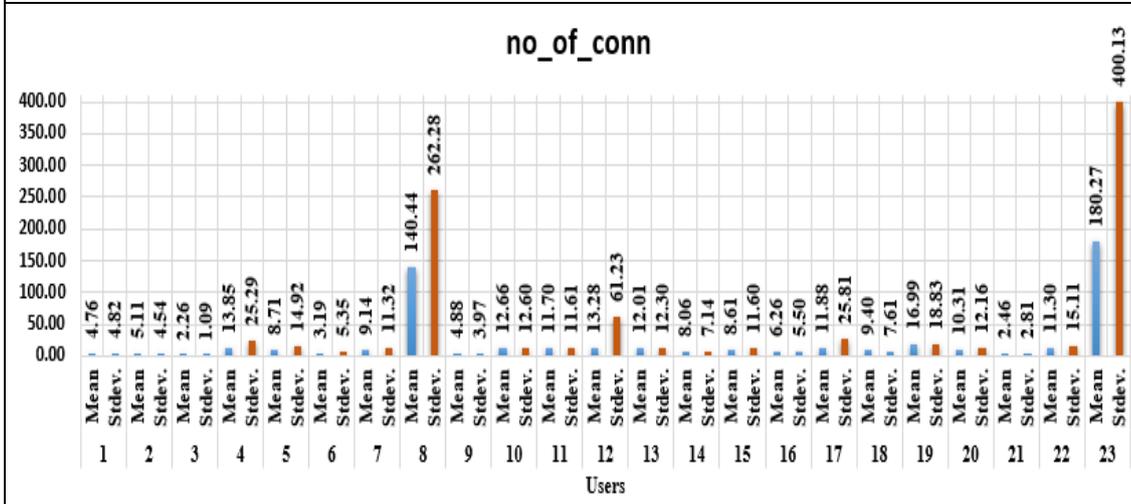
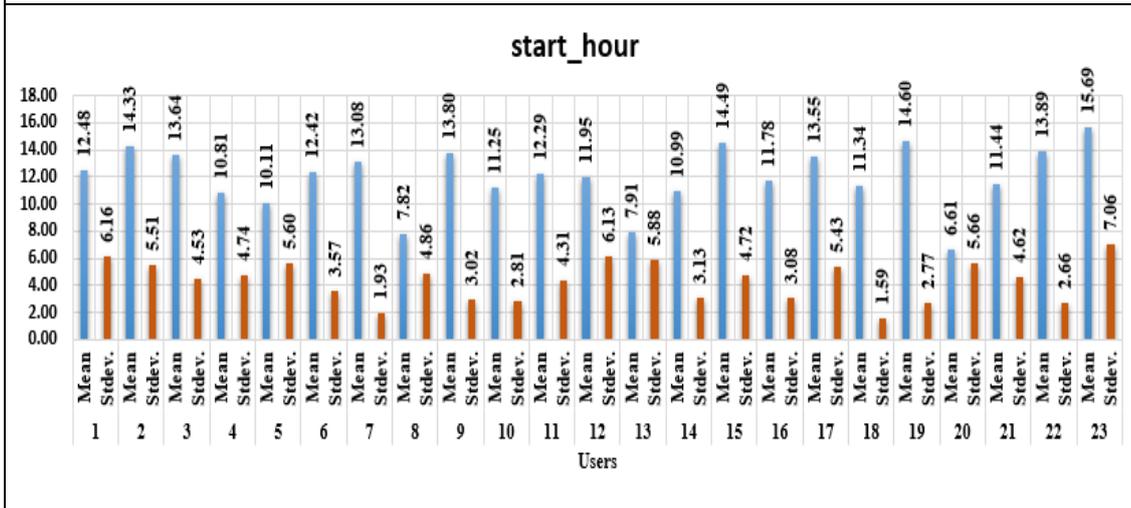
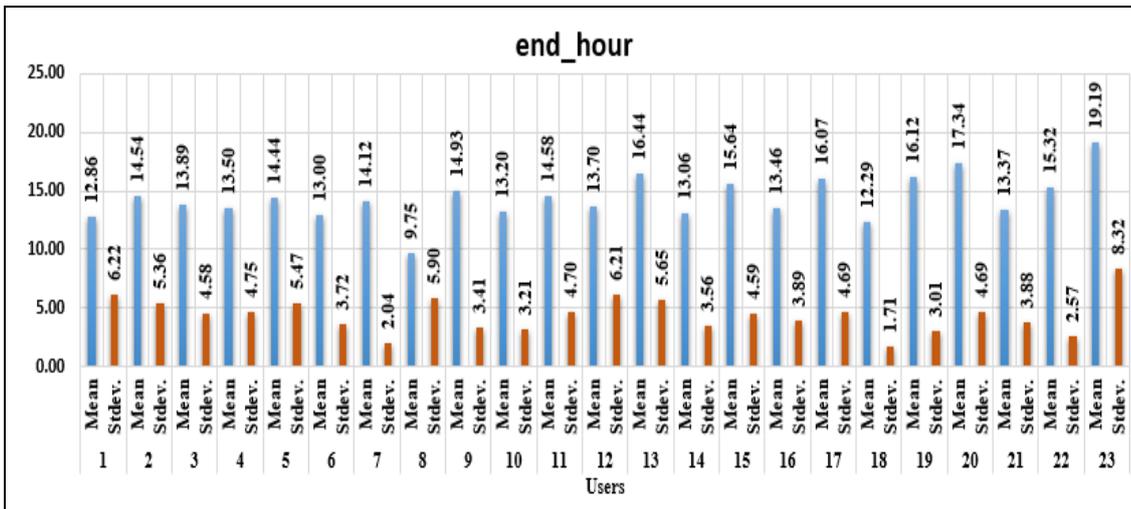
The analysis shows a potential similarity between users 10, 14 and 18 in terms of the mean and SD frequency of the ‘start\_hour’ feature, which affects the performance of these users among the other users in a positive way, as most of these users scored a TPIR with an accuracy of above 77%. Users 14 and 19 showed a different frequency for the mean and SD, which indicates that the users on the system have discriminative information that causes the accuracy to differ when users have different TPIRs on the system.

The 'no\_of\_conn' feature, which represents the amount of data exchanged while the user interacted with the web application, is explained in section 3.2.2. The mean and SD variance shows the close similarity between the frequencies of users 1, 6 and 9, which indicates a similarity in usage between them; users 7 and 13 also share the closest frequency among the users on the system. The similarities and differences between user frequencies indicate that the traffic information related to each user has the ability to enable identification and differentiation between the users on the system.

The analysis of the 'max\_inpkt' feature shows that users 4 and 5 have almost similar frequencies based on the mean and SD variance, which affects the TPIR for these users as they have a close degree of accuracy. In addition, users 1 and 8 have almost the same frequency based on this feature and a close TPIR accuracy.

Upon further analysis across the top five features during the 60-day period, Figure 5.4 below clearly shows the aspects which were previously discussed regarding the mean and SD variance.





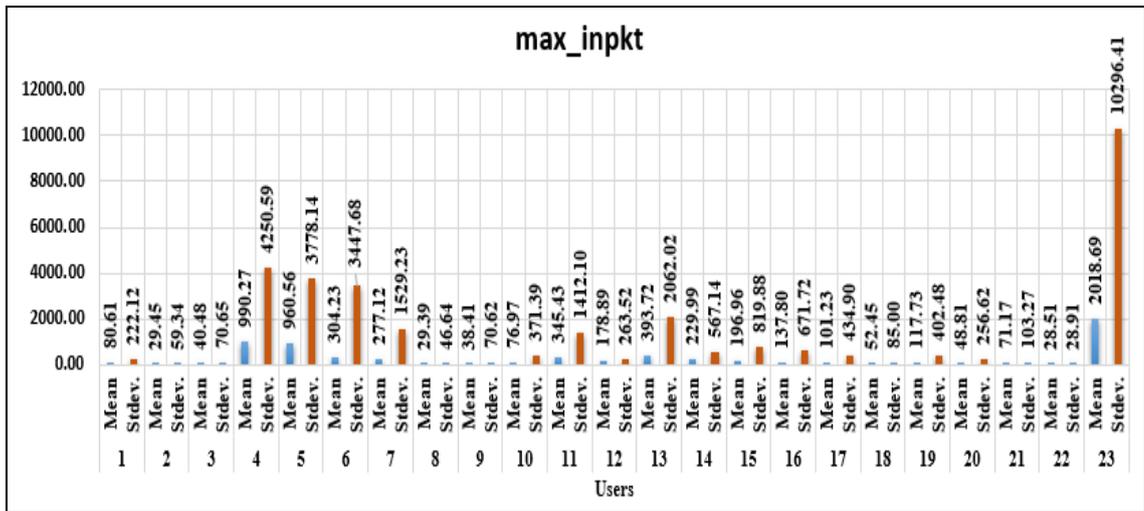


Figure 5.4: Hourly level of top five mean and SD variance

Figure 5.4 illustrates the pattern of usage for each user's mean and SD variance. For instance, for the 'app\_encoded' feature, the SD and mean among the users on the system are very similar to each other. This indicates that the users on the system mostly engaged in similar behaviours. User 1's mean frequency is 2.81 and the SD is 2.57, in comparison with User 2 who scored almost a similar frequency with a mean of 2.58 and an SD of 2.45. The similarities between these two users indicate that their behaviour when interacting with the applications was similar, with a difference of 0.23 for the mean and 0.12 for the SD.

In contrast, User 8's frequency based on the mean and SD is higher than for users 1 and 2, with a mean frequency of 5.03 and an SD of 3.98. This is a difference of 2.22 for the mean frequency compared with User 1, and 2.18 compared with User 2, and the standard deviation difference is 1.41 for User 1 and 1.53 for the standard deviation. Therefore, this variance between users indicates that their behaviour on the application differs, which might affect their performance in the classification. User 23's analysis based on the mean and SD shows that variance in the user's frequencies affected the accuracy of the identification of the individual user. Based on the evaluation of the experiment, User 23 scored the lowest accuracy of all the users on the system, which might be due to his/her

variance frequency based on the mean (6.25) and the SD (7.62); this is the highest frequency among all the users of the 'app\_encoded' feature.

On the other hand, the mean and SD variance of the 'start\_hour' feature frequencies varied between the users on the system. For instance, a comparison between User 1 and User 2 shows that User 1 has a mean frequency of 12.48 and an SD frequency of 6.16, and User 2 has a 14.33 mean frequency and 5.51 for SD frequency. The difference between these two users in mean frequency is 1.85, which suggests that these two users have very close activity behaviour based on one of the timing features during the 24-hour daily activity. The difference between the two users regarding the SD is 0.65, which indicates that their very similar behaviour activity affected their performance on the system. In contrast, based on the mean and SD frequency, User 23 shows a large difference when compared with User 1, as User 23 scored a mean frequency of 15.69 and an SD of 7.06, which differs from User 1 by 3.21 for the mean frequency and 0.9 for the SD. This is high based on the frequencies recorded for this feature. The highest frequency for the mean and SD of User 23 among all the users on the system indicates that User 23's behavioural activity varied, with a high frequency between other users on the system. Therefore, when compared with the analysis of the accuracy results, User 23 scored the lowest accuracy of all the users on the system.

The mean and SD analysis for the 'end\_hour' feature varied between users, which indicates the variances in behaviour and usage between the users on the system. For instance, the frequency of the 'end\_hour' feature means and SDs for User 1 and User 2 is almost the same, which indicates a similarity in their behaviour and usage. User 1 scored a frequency of 12.86 based on the mean and 6.22 based on the SD frequency. User 2 scored a frequency of 14.54 based on the mean and 5.36 based on the SD. The differences between User 1 and User 2 are 1.68 for the mean and 0.86 for the SD, which indicates the

similarity between these two users regarding their daily behaviour and activity, as the frequency difference is not very high between them. In comparison, the mean and SD frequencies for the 'start\_hour' and 'end\_hour' features differ slightly between User 1 and User 2, which gives another indication of the similarity of usage between these two users. The big variance seen between User 23's frequencies and those of the other users on the system shows that User 23's behaviour and activity differed based on the 'end\_hour' feature frequency. The different behaviour of User 23 affected the accuracy of his/her performance, as demonstrated by the results, discussed in section 5.2.1, because User 23 scored the lowest accuracy on the system.

The 'no\_of\_conn' feature, which represents the daily amount of data exchanged and the number of connections between users, was one of the top five feature scores on the system for identifying a user based on the feature importance analysis of the features discussed. The 'no\_of\_conn' feature scored similar frequencies based on the mean and SD analysis among most of the users on the system, except for users 8 and 23. User 8 scored a frequency of 140.44 for the mean and 262.28 for the SD, which indicates that the behaviour of this user based on the 'no\_of\_conn' varied, with large frequencies between other users on the system. When comparing User 8 and User 9 based on frequencies, User 9 scored a frequency of 4.88 for the mean and 3.97 for the SD. The difference in the frequencies, with a score of 135.56 for the mean and 258.31 for the SD, indicates a variation between User 8 and User 9 in their behaviour and activity based on the 'no\_of\_conn' feature. Based on the results discussed previously, User 23 scored the lowest accuracy on the system. The frequency variance in the behaviour of User 23 in having the highest mean and SD among all the users shows that this large variance affected User 23's performance on the system.

According to the mean and SD descriptive analysis, the 'max\_inpkt' feature varies between users on the system. The frequencies of the mean and SD indicate fairly similar behaviour among the users on the system, except for users 4, 5, 6 and 23. User 4 scored a frequency of 990.27 for the mean and 4250.59 for the SD, which is a difference from User 5 of 29.71 based on the mean. Therefore, the fairly small difference between these users indicates similarities between their behaviour. In addition, the difference in SD between User 4 and User 5 is 472.45, which shows that their behaviour and activity are slightly similar based on the 'max\_inpkt' feature. User 6 scored a frequency of 304.23 based on the mean and 3447.68 based on the SD. In comparison, User 23 scored a mean frequency of 2018.69, which is higher than User 6 by 1741.57, which highlights the large variance in User 23's behaviour and activity. User 23's identification rate was very low among all the users on the system, as discussed on 5.2.2, as demonstrated by high variance in the mean and SD between User 23's behaviour and the other users on the system, according to the 'max\_inpkt' feature analysis.

The above analysis discussed the variance and usage patterns illustrated in Figure 5.4. The analysis shows an inverse relationship between the mean and SD frequencies and TPIR accuracy for each user. The inverse relationship shows that when the mean and SD increased, the accuracy decreased in most cases. For instance, User 23 had the highest frequency but the lowest TPIR among all the users for almost all the analysed features, which demonstrates the inverse relationship between the accuracy, mean and SD shown in the descriptive analysis of the frequencies.

### **5.3 A novel feature set for user identification and profiling using application-level quarter-hour session timing features**

This section presents the results of the second experiment, which is based on the flow sessions and quarter-hour session timing resolution features explained in section 3.2.2. The dataset of quarter-hour session timing resolution features, which contains 172

features, was implemented with different flow session thresholds (6, 12, 18, 24, 30 and 36 sec) and produced five sets. Each set was divided into 70% for training and 30% for testing as an unseen dataset to be applied to the gradient boosting classifier separately. The datasets were implemented with different flow inter-arrival time thresholds in order to compare the performance of the five sets. As mentioned throughout the conduct of the experiment, when using a different dataset with a different threshold, the results differed, which suggests that the threshold set is one of the parameters that affected the proposed timing features. The results show that the accuracy varied between the five sets based on the quarter-hour timing features and the different thresholds for the sessions increased the accuracy for the whole population and for individual users on the system.

In contrast, the quarter-hour timing features scored higher accuracy than the hourly timing features in general, as discussed in section 5.2. The shorter, quarter-hour time slots positively affected the system compared with the longer time slot hours, as demonstrated by this experiment. The highest accuracy recorded among the five sets was achieved by using a dataset implemented with quarter-hour timing features and a 30 sec flow inter-arrival time threshold between sessions, as explained in detail in the next section. The results also show that the lowest accuracy accrued with a 6 sec flow inter-arrival time threshold between sessions, which means the flow inter-arrival time threshold between sessions might affect the extracted quarter-hour timing features. The influence changed between the five sets based on the results scored for the whole population on the system, which indicates the variance between the five sets implemented in the quarter-hour timing features experiment. That this experiment scored results higher than those for the hourly flow session timing resolution experiment might be due to several reasons. For example, the different number of features, number of samples, different threshold values implemented for the timing features, and the size of the time unit tested, which was from

0-23 in the hourly timing feature experiment and increased to 0-95 for the quarter-hour timing features.

### **5.3.1 User identification rates: experimental results**

The results show that the quarter-hour timing features and extracted statistical features analysis produced a notably better result in terms of user identification and behaviour profiling when compared with the hourly timing features dataset. The experimental results achieved a high TPIR accuracy, ranging from 62% to 94% accuracy between all ranks. In this experiment, five sets were implemented with quarter-hour timing features, as explained in 3.2.2, and different threshold values (6, 12, 18, 24, 30 and 36 sec). The reason for implementing different thresholds was to assess the effect of the proposed session method and the proposed quarter-hour timing feature with different threshold values on the system, and to what extent the TPIR would be affected by the users' ranks based on the five sets implemented.

The hourly timing feature experiment sets discussed in section 5.2.1 were implemented with a 10 sec threshold, which scored a reasonable level of accuracy for the whole population and for individual users. The quarter-hour timing feature sets were implemented with different threshold values to investigate the level of efficiency when changing the threshold values with the proposed timing features to identify users. The accuracy of set 5, which represents the quarter-hour timing resolution and a 30 sec threshold, exceeded the accuracy of set 1, set 2, set 3 and set 4, which represented other datasets with different thresholds. Set 5 scored a high level of TPIR accuracy from rank 1 at 78%; this value gradually increased to 81% in rank 2, 85% in rank 3 and 90% in rank 4 and continued to improve until achieving 94% TPIR accuracy in rank 5. This means that the system identified users and produced accurate results. Set 6 also showed a good

level of accuracy (74% in rank 1), which increased to 75% in rank 2 and continued to improve until reaching 91% in rank 5.

In addition, some sets scored a low TPIR, such as set 3 and set 4, whereby the system correctly identified more than 70% of set 3 in rank 1 and improved until reaching 89% TPIR accuracy in rank 5. In addition, set 4 scored 73% in rank 1 and continued to improve up to 90% in rank 5. Set 1 and set 2 had the lowest level of accuracy, whereby set 1 TPIR was 62% in rank 1 and improved up to 84%, and set 2 improved slightly when compared with set 1 TPIR, with 66% in rank 1 and reaching 87% in rank 5. Rank 1 accuracy significantly increased from set 1 to set 5. Set 1 scored 62% accuracy compared with set 2, with an increase of 4%. In addition, the accuracy of set 3 increased significantly by 8% compared with set 2, which indicates discrimination in user information. The accuracy of rank 1 increased accordingly between sets until reaching the highest accuracy recorded, based on set 5 increasing by 12%, which indicates the effect of the proposed quarter-hour features and the variance in users' behaviour, which enhanced the performance of the system. The accuracy of set 6 in rank 1 decreased by 4% compared with set 5, which scored the highest accuracy for the whole population in the quarter-hour timing experiment.

The results for some sets attained enough discriminative information to be correctly identified in ranks 1 and 2, such as set 5 and set 6. The remaining sets almost scored a significant improvement in their performance from rank 1 to 5 of between 5% to 20% (set 1, set 2, set 3 and set 4), as illustrated in Table 5.6. The five sets implemented with quarter-hour timing features demonstrated the ability of the system to identify users. The results for the whole population vary between sets based on the behaviour activity of the users on the system. Observation of the samples in the set shows the application usage variations between users based on the quarter-hour timing features. For instance, User 1

might use Facebook differently from User 2, which affects the volume of traffic when comparing the two users and the number of quarter-hour timing slots based on user behaviour. The variance in usage behaviour might influence the performance of the users on the system, which might affect the performance of the five sets, as discussed in the previous paragraph.

Therefore, it can be seen that different influences affected the accuracy when comparing this study and previous works (McDowell, 2013; Vinupaul *et al.*, 2017), such as volume of traffic and the environment in which the flow network traffic was collected. For example, McDowell (2013) achieved an accuracy of 59.7% for user identification by using network traffic collected from a government office with heavy use workers, whereas the quarter-hour timing features in this experiment achieved an accuracy of 78-94%. On the other hand, Vinupaul *et al.* (2017) achieved an accuracy of 83% for user identification by using a dataset collected from a residential home users' environment, which affected their accuracy compared with the accuracy scores based on the quarter-hour timing features and different data collection environment in this study. Therefore, set 1, set 2, set 3, set 4 and set 6 were affected by the lower number of samples between the sets, as the session threshold changed between the different datasets, which affected the level of accuracy between the users' TPIR ranks. The highest accuracy for set 5 improved the proposed measurement features that had been affected by the periods of data collection and user access limitations, as observed in the volume of traffic data between all the sets.

**Table 5.6: Quarter-hour features – users' traffic identification rate results**

Sets	Threshold	Rank1	Rank2	Rank3	Rank4	Rank5
Set 1	6	62%	70%	77%	83%	84%
Set 2	12	66%	75%	80%	83%	87%
Set 3	18	70%	76%	79%	85%	89%
Set 4	24	73%	79%	83%	86%	90%

Set 5	30	78%	81%	85%	90%	94%
Set 6	36	74%	75%	81%	84%	91%

### 5.3.2 Individual user TPIR ranks

At the level of the individual user, there were a number of promising results shown by the scores across users based on set 5, which scored the highest accuracy based on the whole population analysis, as discussed in section 5.3.1. Table 5.7 illustrates the TPIR ranks to show the ability of the proposed feature set to identify and profile users based on application-level sessions. The set 5 features set was implemented based on test data to show the TPIR rate for each user on the proposed system in relation to the probability ranks for user samples to investigate the performance of the system based at the individual user level. The classification model recorded high accuracy for all users, ranging from 45-100%, to identify users' profiles. The labels indicated the users' ID (from User 1 to User 23), as illustrated for each user profile TPIR rate. User 21 scored the highest TPIR from rank 1 with 96% and this continued to 100% in rank 5. This suggests that the system correctly classified almost all his/her samples by assigning them to his/her profile. User 18 scored the second-highest TPIR accuracy in rank 1 among all the users (93%) and this continued to improve until reaching 100% in rank 5. However, comparison of the TPIR ranks between User 18 and User 21 show that for User 18 and rank 2 the TPIR accuracy was higher than for User 21, which indicates the dissemination of the proposed feature on the system.

User 3 recorded the third-highest score on the module with 89% TPIR rank 1 accuracy, which continued to increase until reaching 99% in rank 5. This indicates that the majority of users scored promising results for identifying user samples and assigning profiles. User 4 recorded the lowest accuracy with 45% TPIR rank 1 accuracy, due to the nature of his/her sample's signature and variance. Figure 5.5 illustrates the TPIR accuracy frequency between users based on individual user analysis. The previous discussion

illustrates the results for the highest and lowest accuracy based on the individual user between identification ranks.

The analysis show a number of promising results from rank 1. Among the 23 users examined, 11 acquired uniqueness of information in their traffic, which contributes to the system making the right classification and achieving a level of accuracy of 80% and more in rank 1. The remaining users, who represent about half, acquired a fluctuating TPIR in rank 1, as illustrated in Figure 5.5. Therefore, in a comparison between users based on rank 1, User 21 scored the highest accuracy, which differs from User 4 by 51% and indicates a large variance in behaviour between these two users. On the other hand, User 18 scored 93% and the second-highest accuracy on the system based on rank 1, which differs by 48% from User 4, indicating a large variance between User 18 and User 4 in their behaviour. User 21 and User 18 showed a difference of 3% in their accuracy scores, which indicates similarities in their behaviour as the system identified them accurately from among the other users on the system. The system identified some users with a similar accuracy based on rank 1, such as User 3 and User 5, with an accuracy level of 89%, which indicates the similarity in behaviour between these users. Furthermore, the identification results show that some of the users had much closer scores for accuracy, which indicates the similarity between these users; for example, the system identified User 1 and User 17 with an accuracy of 83% and 84%, respectively.

**Table 5.7: Users' TPIR ranks for set 5 using quarter-hour timing features**

User	Rank1	Rank2	Rank3	Rank4	Rank5
1	83%	87%	89%	90%	90%
2	75%	77%	83%	87%	94%
3	89%	90%	93%	93%	99%
4	45%	60%	71%	76%	88%
5	89%	91%	94%	95%	97%

6	84%	88%	92%	95%	98%
7	71%	74%	78%	81%	94%
8	76%	80%	83%	87%	89%
9	86%	86%	89%	92%	98%
10	76%	82%	86%	93%	96%
11	84%	87%	90%	94%	94%
12	78%	79%	83%	86%	90%
13	68%	70%	76%	84%	89%
14	80%	83%	88%	90%	96%
15	65%	71%	75%	86%	93%
16	85%	87%	87%	89%	87%
17	82%	84%	87%	92%	91%
18	93%	96%	98%	99%	100%
19	68%	70%	77%	87%	92%
20	76%	81%	87%	95%	97%
21	96%	96%	96%	98%	100%
22	67%	72%	76%	83%	89%
23	70%	74%	76%	89%	91%

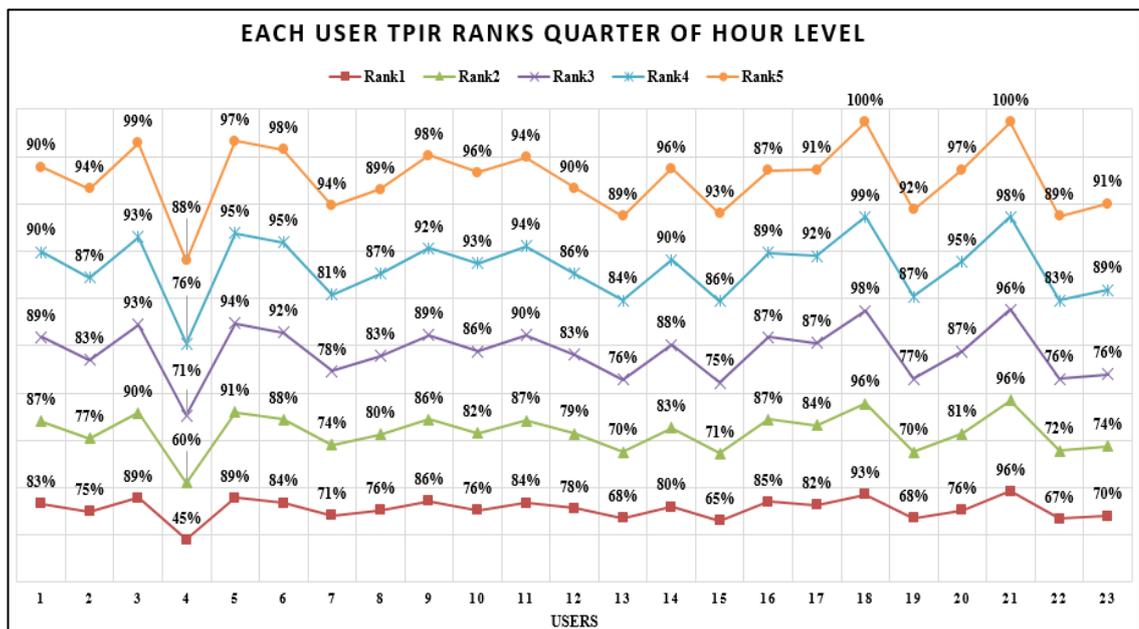


Figure 5.5: Users' TPIR ranks for quarter-hour timing features

As indicated above, the system achieved promising results. Some users did not have sufficient uniqueness of information in their traffic to contribute to the system being able to discover a pattern in terms of user behaviour in a particular time slot in order to profile

them, hence the TPIR was quite low. However, the system identified half the users accurately with a level of accuracy of 70% and over, which indicates discriminative information in the user traffic based on the proposed timing features, which enhanced the system's ability to identify users with an accurate TPIR.

### **5.3.3 Confusion matrix**

This section presents the analysis of the set 5 dataset from different perspectives based on individual user implementation using a confusion matrix and the precision, recall and F1 score measurements and equations set out in section 5.1. Table 5.8 shows the confusion matrix for all users to demonstrate the correct and incorrect prediction of each class based on the test data for set 5. The performance of the classification model is high for all the classes, ranging from 70-96%. The labels indicate the users' ID (from User 1 to User 23), as illustrated in Table 5.8 for the predicted and true labels. The highest score for the actual class predicted was 96% as the TP for User 21 and the FN was 4% misclassified when recorded for User 11, which indicates the ability of the module to identify users with a high score.

Furthermore, User 18 scored the second-highest accuracy from among all the users (93% of TP classified samples) but 7% misclassified (FNs) were recorded for users 4, 12 and 21. In addition, User 5 recorded the third-highest score on the module with 89% TP, and 11% misclassified attributed to users 2, 9 and 11. User 4 recorded the lowest accuracy with 45% TP, and 55% misclassified attributes for users 1, 3, 7, 14, 15, 16 and 19 on the module because of the number of traffic samples being the lowest among all the users on the module. The reason for the low accuracy score for User 4's traffic samples was the number of days, which affected the TP for this user when compared with the other users on the system. Observation of User 4's traffic samples shows that there are about 17 days

without any activity by this user over the whole 60-day period, which is lower than for the other users on the module.

Another factor affecting the accuracy of User 4 is the variance in feature frequencies between User 4 and the other users on the module based on the SD and mean. The descriptive analysis based on the mean and SD for the top five features on the system for each user shows that User 4 scored the highest frequency, which indicates the variance in behaviour between User 4 and the other users on the system. For instance, at the 'end\_quarter\_of\_an\_hour' feature descriptive analysis, User 4 scored the highest frequency of all the users on the system investigated using quarter-hour timing features: 76.66 for the mean frequency and 30.13 for the SD frequency. This indicates the variance between User 4 and the other users on this system (more details are provided in section 5.3.5).

**Table 5.8: Quarter-hour timing features confusion matrix (features set 5)**

U <sub>id</sub>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	83	0	0	5	0	0	0	0	3	0	0	0	0	0	5	4	0	0	0	0	0	0	0
2	5	75	0	0	8	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
3	0	0	89	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	5
4	4	0	8	45	0	0	9	0	0	0	0	0	0	7	8	10	0	0	9	0	0	0	0
5	0	2	0	0	89	0	0	0	5	0	4	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	84	0	0	11	0	0	0	0	0	0	0	0	0	0	0	5	0	0
7	7	0	10	0	0	0	71	0	0	0	4	0	0	0	0	0	0	8	0	0	0	0	0
8	0	0	0	0	8	0	0	76	0	0	0	9	0	0	0	0	5	0	0	0	0	2	0
9	0	0	0	4	0	0	0	0	86	0	0	0	0	0	0	0	0	0	0	0	5	0	5
10	0	6	0	0	0	0	5	0	0	76	0	0	0	0	6	0	0	5	2	0	0	0	0
11	5	0	0	0	0	3	0	1	0	0	84	0	0	0	5	0	0	0	0	0	2	0	0
12	0	0	4	0	0	0	0	0	0	3	0	78	0	0	0	0	9	0	0	0	0	6	0
13	0	0	2	0	0	6	0	4	0	0	0	4	68	4	4	0	0	4	0	4	0	0	0
14	0	6	0	0	4	0	0	0	0	0	0	0	6	80	0	0	4	0	0	0	0	0	0
15	6	0	0	0	0	0	1	0	0	0	0	0	0	0	65	0	0	10	0	4	6	0	8
16	0	0	0	0	0	0	0	0	0	5	0	0	5	0	0	85	0	0	0	0	0	5	0
17	0	5	0	5	0	0	0	1	0	0	0	0	0	0	0	0	82	0	7	0	0	0	0
18	0	0	0	2	0	0	0	0	0	0	0	3	0	0	0	0	0	93	0	0	2	0	0
19	0	0	0	1	0	9	0	0	7	0	0	0	0	8	0	0	6	0	68	0	0	0	10
20	0	0	4	0	0	0	0	0	0	5	0	0	0	0	10	0	0	0	76	0	5	0	0
21	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	96	0	0
22	0	0	0	0	6	0	3	0	0	0	0	9	0	0	8	0	0	0	0	7	0	67	0
23	1	0	0	0	0	0	0	3	0	0	12	0	4	0	0	0	0	10	0	0	0	0	70

The previous section explored implementing TPIR metrics and ranks to assess the accuracy for the whole population, as well as individual users, as discussed in section

5.3.1. It was demonstrated that the system has the ability to identify users with a high degree of accuracy. Therefore, this section presents the analysis to extend the rank 1 accuracy for each user based on the information the confusion matrix provided. The confusion matrix has the ability to extract the TP, FP and FN, which are the most important parameters for measuring the recall, precision and F1 score.

Figure 5.6 shows the analysis and results of the different types of measurement and the calculations relating to each user using the equations explained in section 5.1, which required information from the confusion matrix table. The results show that User 1 scored 83% recall and 74% precision, which indicates that the system identified this user with a good score. When recall is higher than precision, this means the TP and FP are higher than the FN, which is an indication that the system has identified this user with a high score. User 2 scored 79% precision and 75% recall, which indicates that the system identified this user with a reasonable score, as the precision is higher than the recall. This means that the FN is higher than the TP and FP. User 3 scored 89% recall and 76% precision, which indicates that the user was identified with a good score. User 4 scored 45% for recall and 72% for precision, revealing a large gap between the two scores and indicates that the FN is very high. This makes this user the lowest-scoring that the system identified for the reasons indicated earlier in this chapter.

As demonstrated above, the calculation of precision and recall affected the score for each user when the recall was high and precision low, and vice versa. In comparison, the TPIR accuracy was implemented together with the confusion matrix metrics (recall, precision and F1 score) analysed in this section. Figure 5.6 presents the variance between users and an analysis of the lowest and highest TPIR accuracy that the system scored for user identification, as the confusion matrix shows the misclassification between users. Therefore, User 21 scored the highest TPIR accuracy among all the users on the system,

as demonstrated here, as they scored a recall accuracy that was higher than the accuracy for precision, which means that the TP and FP are higher than the FN. The higher score for recall when compared with precision indicates that the system accurately identified User 21. In comparison, User 13 scored higher for precision than for recall, which means that the FN is higher than the TP and FP. From this can be derived that User 13 scored a reasonable level of TPIR accuracy.

On the other hand, User 4 scored the lowest TPIR accuracy on the system but the accuracy of the precision is higher than for recall, which means that the FN is higher than the TP and FP; this indicates that most of User 4's samples were assigned to other users on the system. Hence, high FN was an effect of the performance of this user on the system. In comparison, User 5 scored an accuracy for recall that was higher than for precision, which enhanced the system's ability to identify User 5 with an accuracy of 89%; this was affected by the high level of TP and FP compared with FN. The higher TP and FP for User 5's samples encouraged the system to identify User 5 with a more accurate TPIR compared with User 4's accuracies. As a result, the confusion matrix and important metrics derived from this information showed the highest and lowest accuracy scored for each user based on the TPIR analysis. The variance in accuracy between all the users' scores based on the average between precision and recall shows the high and low accuracy scored for each user based on the TPIR analysis. The link between this section and the TPIR is that if the TP and FP are higher than the FN, the recall is higher than the precision, which means that in most cases the system identified the user with accuracy. In contrast, if the FN is higher than the TP and FP, the precision is higher than the recall, which means that in most cases (as shown in this section) the system identified the user with a reasonable or low TPIR.

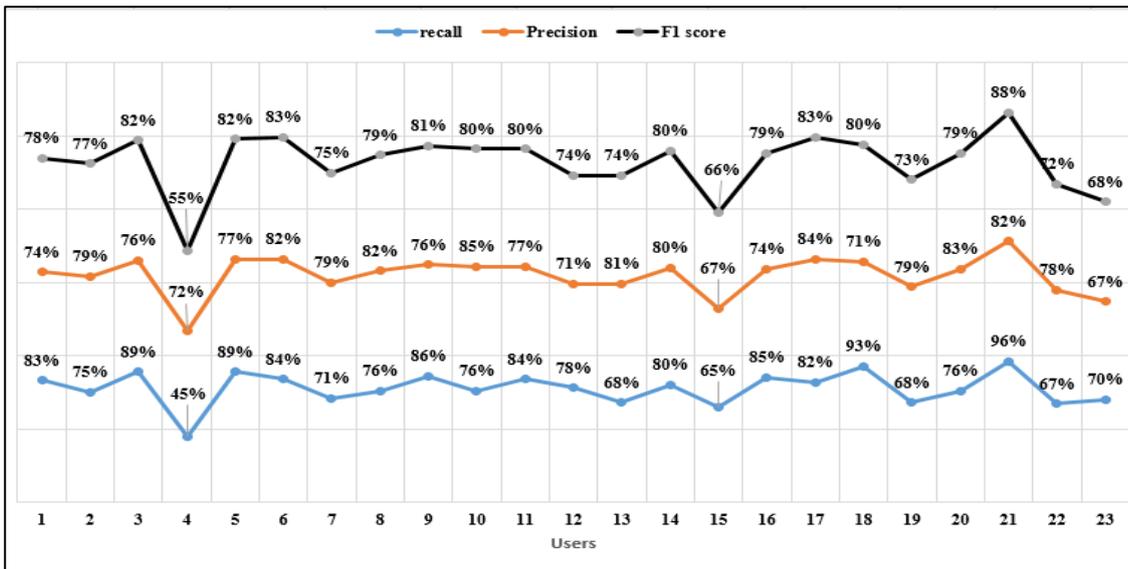


Figure 5.6: Quarter-hour for individual users (recall, precision and F1 score)

### 5.3.4 Feature importance

The importance of the features in this experiment was explored by analysing the data using the random forest ranking technique. Feature importance analysis provides a method for assessing the influence of the proposed quarter-hour timing and the statistical features extracted by ranking the features from most important to less important on the system. The random forest ranking technique provided the weights and ranking by applying the set 5 dataset discussed in section 5.3.1 to identify the relative importance of the features in this experiment; moreover, 172 features have been extracted, as explained in section 3.2.2.

The top 40 features are presented according to weight in Table 5.9, in which the top three features (app\_encoded, end\_quarter\_of\_hour, and number\_of\_connections) scored the highest usage between the extracted statistical and proposed quarter-hour session timing features. The features importance analysis ranked the proposed timing resolution features according to the highest usage of the features on the system because the proposed timing features enhanced the classifier to identify and discriminate users. The highest-ranking weight achieved between the proposed features was recorded for the ‘app\_encoded’

feature, with a weight of 0.030984 and a difference of 0.00422 from 'end\_quarter\_of\_hour', as shown in Figure 5.7. As the 'end\_quarter\_of\_hour' feature is ranked with a weight of 0.02676, this makes it the second-highest ranked among the 172 features. The 'no\_of\_conn' feature ranked the third-highest weight on the system with a weight of 0.021528, which differs from 'sum\_received\_to\_transmitted' by 0.000569, which is ranked fourth.

In addition, the 'start\_quarter\_of\_hour' feature scored the fifth-highest rank on the system with a difference of 0.00617, which indicates the discriminative information provided by the proposed timing features on the system. The 'app\_encoded' feature enhanced the ability of the module to identify users, which indicates that the application used plays an important role in differentiating between users' behaviour, as demonstrated by the highest rank recorded for this feature. Therefore, the 'app\_encoded' feature was stored as an encoded integer feature in the dataset, as discussed in 3.2.2, to investigate the influence of the application when used by a user and to what extent the variance between the applications used affects the identification system. The 'end\_quarter\_of\_hour' feature scored the second-highest of all features, which highlights the importance of the quarter-hour timing resolution feature in identifying users from their investigated traffic samples, as the 'end\_quarter\_of\_hour' feature represents the encoding time of the end usage of each application used by the user. In addition, the 'start\_quarter\_of\_hour' feature score showed it to be one of the top five features on the system, as it represents the start of the quarter-hour time of each application used by the user.

The 'start\_quarter\_of\_hour' and 'end\_quarter\_of\_hour' features were stored on the dataset as encoded values from 0-95 to represent user activity based on the application used, which enhanced the system's performance as the ranking weights of the feature importance analysis scored them with the top highest ranks on the system. The importance

of the proposed features not only includes the timing encoded features, but also the 'no\_of\_conn' feature, which represents the amount of data exchanged in the application connection of each sample. Hence, the high rank recorded for the number of connections feature demonstrates the amount of data exchanged when a user interacts with the application and enhanced the ability of the system to identify and differentiate between users. For instance, the amount of connection data exchanged might vary between User x and User y in terms of using Facebook and also vary between them when using Instagram.

As a result, the proposed method and the timing features extracted and encoded encouraged the system to identify users and discriminate their traffic in a way that is clearly visible in Figure 5.7, in that the proposed timing features are ranked as the top three features on the system. In addition to the 'quarter\_of\_hour' session timing resolution features, the feature numbers (4, 7, 8 and 9) that were calculated and proposed ranked a high score among the features and enhanced the module's ability to identify users on the system.

It can be seen that there is a small difference in the feature importance values between the hours of the day and the quarter-hour feature sets. This is because the number of features (vectors) in each of the sets is not the same: there are 101 features in the case of hours (24 hours plus other statistical features) and 172 features (96 quarter hours plus other statistical features) in the case of a quarter-hour features set. The resolution in these timing-based features also has an influence on the classification decision. For example, let us assume that one user starts transmitting network traffic (checking work emails) every day at 9:14-9:15 am, while another user starts the same process at 9:44-9:45 am. This also assumes that they both use Microsoft Outlook to access email. In terms of timing pattern, they start every day at the same hour but they differ in terms of the quarter hour. The first user can be better identified using the quarter hour, as the second user has yet to

start transmitting traffic two quarter hours later. Therefore, the first user's quarter hours could have a more significant value than using hours when it comes to classification for this time slot.

**Table 5.9: Quarter-hour timing feature importance weights**

<b>Feature</b>	<b>Ranking weight</b>	<b>Feature</b>	<b>Ranking weight</b>
app_encoded	0.03098	mean_in byte	0.012368
end_quarter_of_hour	0.02676	median_out byte	0.012117
no_of_conn	0.021528	min_in pkt	0.011939
sum_Received_to_transmitted	0.020959	median_bpp	0.011702
start_quarter_of_hour	0.020598	sum_out_pkt	0.011399
min_bpp	0.019643	median_D2S pps	0.011268
min_Received_transmitted_data	0.017577	sum_in_pkt	0.011133
sum_D2S bpp	0.017354	mean_D2S bpp	0.010749
min_D2S bpp	0.017055	mean_bpp	0.010681
sum_bpp	0.015802	sum_D2S pps	0.010439
max_out_pkt	0.015591	min_out byte	0.01043
mean_out_pkt	0.015022	sum_Received_transmitted_data	0.010404
mean_in_pkt	0.014394	min_D2S pps	0.010352
median_in_pkt	0.013215	median_Received_transmitted_data	0.010292
mean_out byte	0.013036	'37'	0.010261
median_in byte	0.012867	median_Transmitted_DR	0.01001
max_in_pkt	0.01285	min_Transmitted_DR	0.009944
median_D2S bpp	0.012813	min_out_pkt	0.009687
median_out_pkt	0.012599	median_bps	0.009603
min_in byte	0.012426	Sum_pps	0.009309

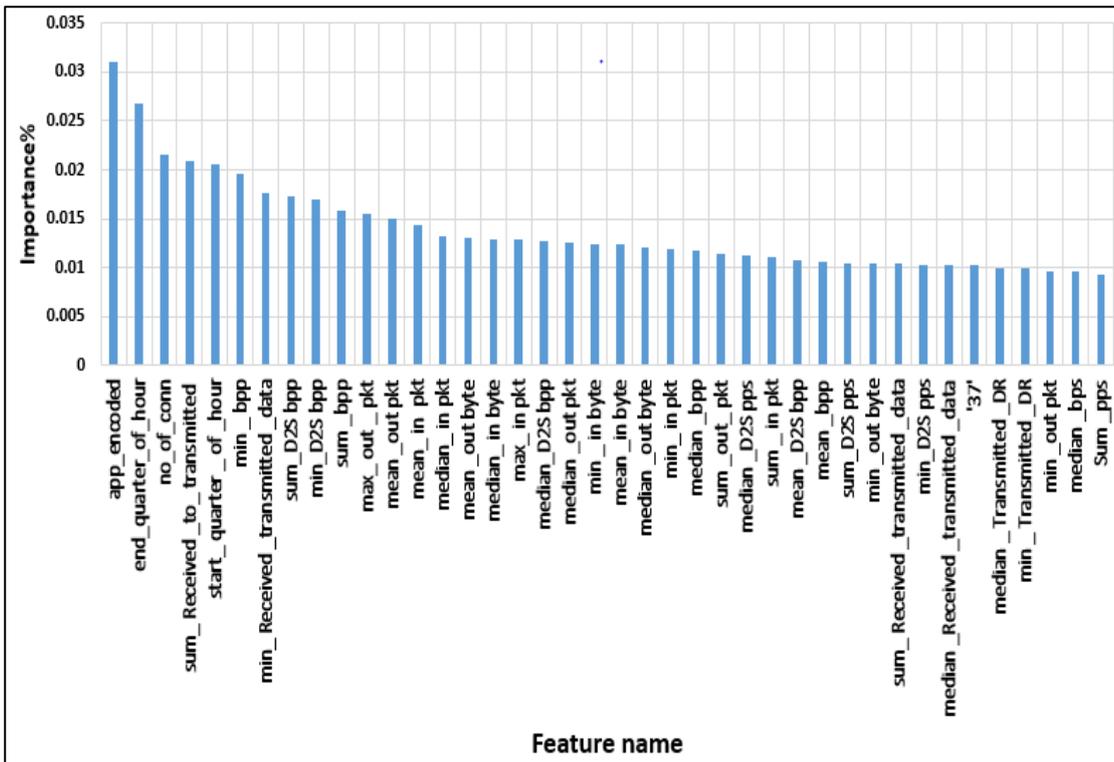


Figure 5.7: Quarter-hour feature importance scores

### 5.3.5 Descriptive analysis of mean and standard deviation variance

A descriptive statistics approach was implemented for the features to identify the individual users' patterns present in the dataset. The classification and performance of the system are affected by the selection of optimum features based on the top features considered in this analytical study. In order to identify the variance in usage among the users' features, Table 5.10 shows the mean and SD for the top five statistically important features on the system for each user, as collected over 60 days to obtain the users' daily usage.

Table 5.10: Quarter-hour top five feature means and SD variances

Users		app_encoded	end_quarter_of_hour	no_of_conn	sum_Received_to_transmitted	min_bpp
1	Mean	2.82	55.82	8.94	9.61	86.36
	SD	2.57	24.65	10.21	10.97	131.47
2	Mean	2.58	61.26	8.73	9.10	88.23
	SD	2.448	21.19	11.06	10.99	86.42
3	Mean	2.11	59.18	2.66	3.01	197.62
	SD	2.331	18.42	1.46	1.65	227.19
4	Mean	5.03	76.66	160.60	170.40	276.40
	SD	3.983	30.13	290.54	304.10	268.95
5	Mean	4.32	63.23	15.07	16.27	114.67
	SD	3.359	22.85	25.88	28.35	134.46

6	Mean	2.63	56.23	4.14	4.73	255.49
	SD	2.81	14.63	8.87	9.71	255.80
7	Mean	4.16	64.14	73.80	77.36	66.65
	SD	3.507	10.17	107.76	115.72	110.54
8	Mean	4.07	41.99	79.84	148.28	78.61
	SD	3.234	23.71	128.13	273.13	63.90
9	Mean	3.13	63.31	6.83	7.61	136.18
	SD	3.175	13.77	6.98	7.47	125.30
10	Mean	4.35	63.72	101.15	107.47	60.01
	SD	3.334	15.43	157.62	168.16	98.59
11	Mean	4.38	70.13	98.24	103.78	97.64
	SD	3.254	18.95	163.96	171.17	88.27
12	Mean	2.5	59.13	13.66	14.42	140.44
	SD	3.139	24.68	61.19	64.39	103.83
13	Mean	2.47	75.26	47.98	49.90	84.92
	SD	3.131	21.80	82.18	87.49	79.41
14	Mean	3.7	70.07	142.23	87.85	66.03
	SD	3.061	17.97	262.86	138.79	108.27
15	Mean	3.19	68.97	48.47	44.51	75.92
	SD	2.705	18.44	83.78	86.69	106.57
16	Mean	4.64	58.60	11.48	12.23	102.92
	SD	3.409	16.91	13.72	15.30	99.56
17	Mean	3.56	75.91	63.64	66.67	70.13
	SD	3.074	19.05	86.91	90.78	105.93
18	Mean	3.56	52.85	14.32	14.92	66.75
	SD	2.988	6.39	13.97	15.17	58.50
19	Mean	4.22	71.30	85.44	90.92	56.41
	SD	3.197	12.84	119.47	127.98	80.54
20	Mean	0.84	62.53	18.22	19.54	127.44
	SD	1.282	20.66	29.84	33.19	102.51
21	Mean	1.14	57.31	2.57	3.04	54.31
	SD	1.503	16.29	3.68	3.82	53.23
22	Mean	3.54	66.44	30.04	34.12	73.78
	SD	2.998	10.28	44.03	53.73	80.32
23	Mean	3.45	55.95	67.04	68.11	59.58
	SD	3.334	8.48	78.77	81.07	39.45

The table above shows that users 1, 2, 3 and 6 shared quite similar frequency in ‘app\_encoded’ mean and SD variance. However, there is considerable difference between these users and the other users on the system. For instance, the mean and SD of users 1 and 2 share a quite similar frequency, which could provide a degree of discrimination for a classifier to identify them. User 1 scored a frequency of 2.82 for the mean, which differs from User 2 by 0.24, and demonstrates the similarity of usage behaviour between these two users. User 1 scored a frequency of 2.57 for the SD, which differs from User 2 by 0.13, giving another indication of the similarity of usage between these two users. In comparison, User 4 scored a frequency of 5.03 for the mean, which differs from User 1 by 2.21 and differs in terms of SD by 1.41. The clear difference between User 1 and User 4 in mean and SD are an indication of the variance in the behaviour and activity of User

4 and the other users on the system. User 4 scored the lowest TPIR accuracy based on the classification analysis discussed in section 5.3.2, which indicates that the high mean and SD frequency of User 4 compared with the other users on the system shows an effect on User 4's performance.

Moreover, User 21 scored a frequency of 1.14 for the mean and 1.503 for the SD frequency, which differs from User 4 by 3.89 based on the mean and 2.48 on the standard deviation. The difference between User 4 and User 21 indicates that the behaviour and activity of these two users varied and affected their performance in the classification analysis. User 21 has the lowest mean and SD frequency for the 'app\_encoded' feature, which improves the user's TPIR in having the highest score among the other users on the system.

However, analysing other features reveals that users 1, 2 and 12 have a fairly similar mean and SD for the 'end\_quarter\_of\_hour' feature. For instance, User 1 scored a frequency of 55.82, which differs from User 2 by 5.44 based on the mean. In addition, User 1 scored a frequency of 24.65, which differs from User 2 by 3.46 and indicates the similarity in behaviour usage between these users. User 4 scored the highest frequency on the system with 76.66 based on the mean and 30.13 based on the SD, which indicates his/her variance with other users on the system. In comparison, User 8 scored the lowest frequency based on the mean and SD, with a difference of 34.67 based on the mean frequency and 6.42 based on the SD. The variance between User 4 and User 8 suggests that the difference between the users' behaviour affected their performance, as, based on the classification analysis, User 4's performance was the lowest among the other users on the system. On the other hand, User 8's performance based on the classification analysis shows a score of reasonable accuracy between the users on the system, which indicates the mean and

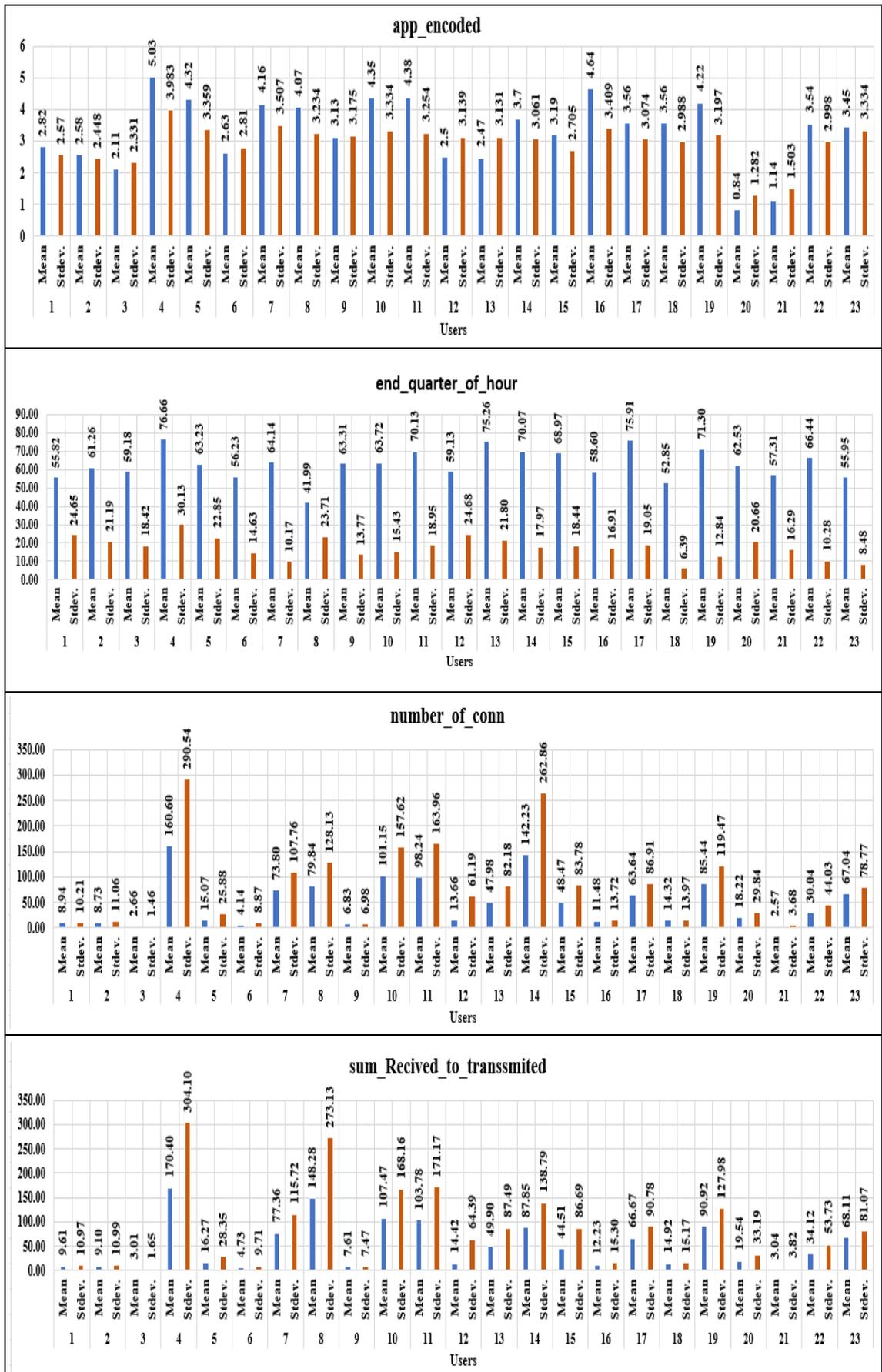
SD variance between the users affected the accuracy results relating to the performance on the system.

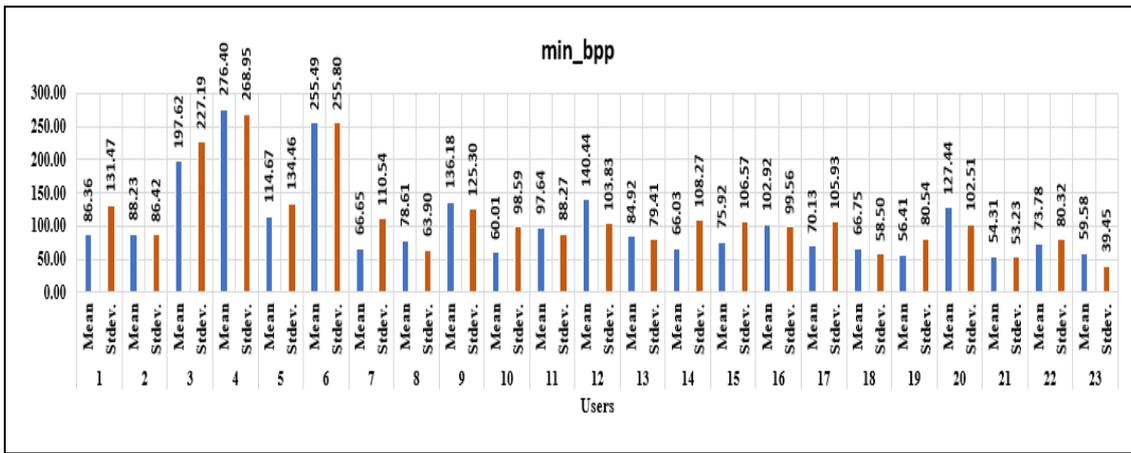
The analysis reveals similarities in behaviour between User 1 and User 2 because they have quite a close level of frequency based on the mean and SD variance of the 'no\_of\_connection' feature. For instance, User 1 scored a frequency of 8.94 based on the mean in comparison with User 2, who scored a frequency of 8.73, with only a small difference between them of 0.21. In addition, User 1 scored a frequency of 10.21 based on the SD and User 2 scored a frequency of 11.06, which differs from User 1 by 0.85. Therefore, the small difference between User 1 and User 2 shows the similarity in the behaviour usage between these users based on the amount of data exchanged for the mean and SD variance. In comparison, User 4 scored the highest frequency among the other users, with 160.60 based on the mean and 290.54 based on the SD; this differs from User 3, who scored the lowest frequency of 157.94 based on the mean and 289.08 based on the SD. The large difference between User 4 and User 3 indicates the variance in the behaviour of these users, which affected their performance on the system, as User 3 scored a reasonable TPIR accuracy and User 4 scored the lowest TPIR accuracy.

The analysis of the 'sum\_Received\_to\_transmitted' feature shows the close similarity between User 1 and User 2 regarding frequency, which indicates the similarity in usage between them, as well as User 16 and User 18, as they share the closest frequency among the other users on the system. For instance, User 1 and User 2 scored a close degree of frequency based on the mean and SD. The difference between User 1 and User 2 based on the mean is 0.06 and 0.02 based on the SD, which demonstrates the similarity in behaviour between these users on the system. User 4 scored the highest frequency based on the mean and SD compared with User 3, who scored the lowest frequency. User 4 scored a frequency of 170.40 based on the mean, which differs from User 3 by 167.39.

Therefore, the SD also revealed a large difference between User 3 and User 4 with a variance of 302.45, which indicates behaviour variance. Based on the mean and SD, the variance between User 3 and User 4 affected their performance on the system. User 4 scored the lowest TPIR accuracy on the system among the users when compared with User 3, who scored a reasonable degree of TPIR accuracy based on the classification analysis discussed in section 5.3.2.

Furthermore, the analysis of the 'min\_bpp' feature shows that the users' mean and SD frequency are different for all the users on the system and there are no significant similarities between them. However, User 21 and User 23 have the lowest frequency among the users on the system, which positively affected the performance of the system. User 21 scored a frequency of 54.13 based on the mean, which differs from User 23 by 5.45. In addition, User 21 scored an SD of 53.23, which differs from User 23 by 13.78 and indicates the similarity between these users' behaviour activity. The similarity between User 21 and User 23 positively affected their performance. Therefore, the TPIR accuracy analysis score shows a reasonable performance for User 21 and User 23 on the system, indicating the similarity between these users and enhanced the performance of these users on the system. In comparison, User 4 scored the highest mean and SD of all the users on the system. The high variance between User 4 and the other users on the system negatively affected User 4's performance, as User 4 scored the lowest accuracy on the system based on the classification analysis applied to evaluate the identification system. User 4 scored a frequency of 276.40 based on the mean, which differs from User 21, who scored a mean of 59.58, with a difference of 216.82, which indicates a large variance between User 4 and User 21 regarding behaviour, and this affected their performance on the system. On the other hand, User 4 has the highest frequency, which means the variance in this user's usage is different from that of the other users on the system and affected his/her accuracy on the system.





**Figure 5.8: Quarter-hour top five features mean and SD variance**

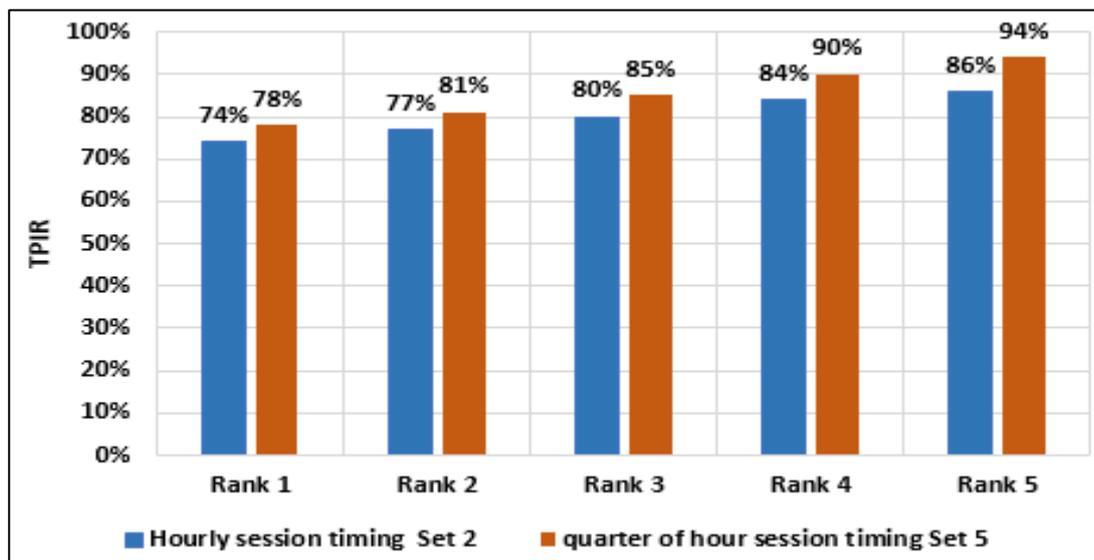
Figure 5.8 illustrates the pattern of usage for each user (mean and SD variance). Upon further analysis across the top five features during the 60-day period, the figure also shows all the aspects discussed previously concerning the mean and SD variance. The mean and SD descriptive analysis of the top five features on the system show the variance in patterns between users on the system. The descriptive analysis shows that some users on the system have a similar behaviour pattern, such as User 1 and User 2, based on most of the features. However, other users on the system have a large variance, which affected their performance on the system based on the accuracy recorded. Based on the mean and SD analysis, in most cases, if the user scored a high frequency based on the mean and SD, this variance usually negatively affected that user's performance on the system, as highlighted in the previous discussion of User 4.

In comparison, the descriptive analysis conducted in section 5.2.5 shows that User 1 and User 2 scored very similar frequencies based on the mean and SD, which indicates the similarity in the behaviours of these users compared with the other users on the system. On the other hand, User 23, who scored the highest frequency in the hourly flow session timing descriptive analysis, had the lowest accuracy on the system. In addition, in the descriptive analysis discussed in this section, based on the quarter-hour flow session timing experiment, User 4 scored the lowest according to the accuracy recorded.

## 5.4 Discussion

The experimental results presented above indicate that the features derived from flow-level generic network traffic are unique for each individual; therefore, using them to identify and build a user behavioural profile is a promising solution to enabling security administrators to make informed decisions about security and enhance their organisation's policy. In addition, the proposed features and the analysis of the user traffic information can enhance user identification and behaviour profiling. Therefore, the TPIR results of the first experiment, which contains the statistical features extracted and the hourly flow session timing features to represent the 24-hour daily user usage activity, indicate that the system has the ability to identify users with accurate results. The highest accuracy recorded by investigating the hourly flow session timing features was based on the following three sets: set 1, set 2 and set 3. Each set contains a different period of time: set 1 represents the first 30 days of the collected data; set 2 represents the last 30 days; and set 3 represents the full 60 days of collected traffic, as discussed in 5.2.1. This facilitated investigating the variance in these periods to attain the most relevant TPIR accuracy. The highest accuracy was scored with the set 2 dataset, which is based on the last 30 days of the traffic collected for the 60-day period examined in this study. The results for set 2 show the ability of the system to identify users with an average TPIR for the whole population of ~86% and for the individual user of ~91%, as explained in section 5.2.1. Moreover, the results of the second experiment show that utilising quarter-hour session timing features and extracting statistical features to represent the 24-hour daily user usage activity based on a shorter timing unit enhanced the system and scored higher accuracy than the hourly session timing features. Therefore, the system was able to identify users based on the application level with a promising TPIR for the whole population of ~94% and for the individual user of ~96%, as explained in section 5.3.1. The set 5 data scored the highest level of accuracy in the second experiment, as these data

represent the proposed quarter-hour session timing features based on a 30 sec flow inter-arrival time threshold, which scored the highest TPIR. However, the other sets, with different thresholds tested (6, 12, 18, 24 and 36 sec), scored a low TPIR accuracy compared with set 5 with a 30 sec threshold, as shown in Figure 5.9. The quarter-hour session timing features scored a higher accuracy of 4% in rank 1 and 8% in rank 5 than the hourly session timing features based on the whole population for the data investigated. The highest accuracy scored for the quarter-hour timing features highlights the clear effect of the timing features in identifying users based on web application usage. Likewise, a significant increase in TPIR was recorded by the system.



**Figure 5.9: Comparison of results for the flow session timing features**

The results above revealed a high level of TPIR accuracy from rank 1, which is a clear indication that there is sufficient discriminating information in the proposed user features to predict accurate identification in the system. The TPIR indicates that the approach provides positive insights in terms of creating user behaviour profiles and the system successfully identified individuals with up to 96% accuracy in the experiments conducted, influenced by the unique information identified among the different users. Figure 5.10 illustrates a comparison of users' TPIR based on the hourly and quarter-hour session

timing features. In general, the performance of users differs between the quarter-hour features for individual users compared with the hourly features. For instance, User 5 shows a significant increase of 31% in rank 1 from the hourly timing features to the quarter-hour features, which indicates that identifying User 5 based on quarter-hour timing was more accurate than doing so using hourly timing. In contrast, User 10's TPIR rank 1 accuracy increased by 21% due to the descriptive information in the hourly timing feature samples compared with the quarter-hour timing samples, as the number of samples increased the accuracy for this user. Hence, the performance of the users showed a significant increase with quarter-hour timing identification compared with using hours, as the increase in most cases was not less than 20% for the TPIR. This study highlights the clear effect of timing on being able to identify users based on application-level usage during the course of 24 hours, as the accuracy varied when using the time slots between hours and quarter hours, as investigated in the experiments.

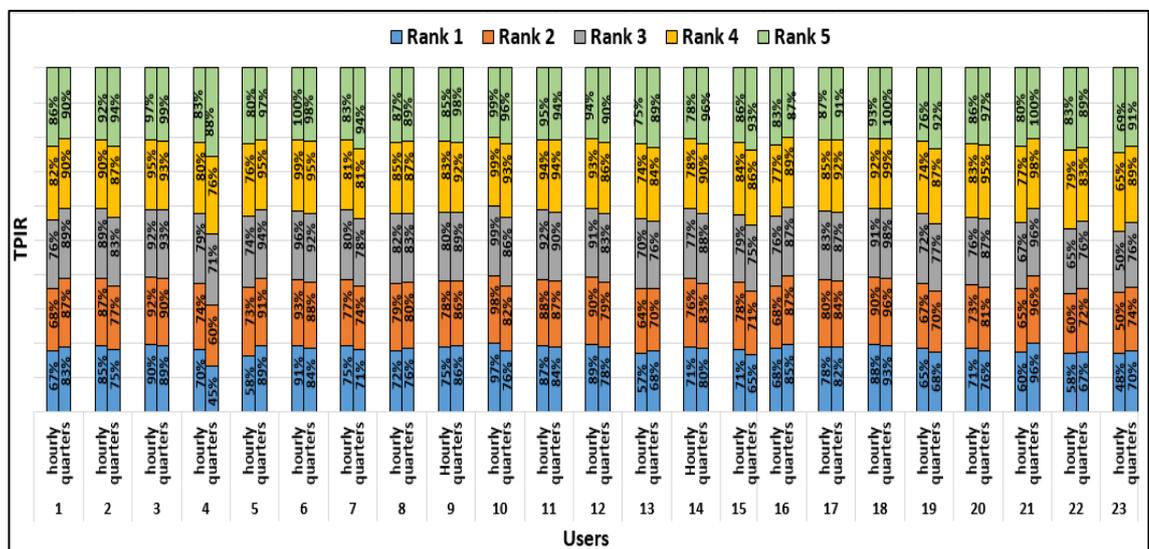


Figure 5.10: Individual user TPIR comparison

As a result, the proposed method, which is based on a flow-level statistical approach, improved the ability to identify users with a high degree of accuracy and minimised the disadvantages of other approaches used to classify traffic, i.e., the problem of dynamic

change and the encryption of traffic. A behaviour profile will help network administrators to monitor and analyse traffic in order to identify users and detect suspicious activity on the network, as well as to make informed decisions for security and policy purposes. The accuracy differed between the users due to the volume of traffic and the period in which the data were collected. The explanation for the predicted label between users being high can be attributed to the level of uniqueness of particular users when using a particular application.

## **5.5 Conclusion**

This chapter has shown that it is possible to identify users based on a set of timing features extracted from traffic metadata. This chapter presented an analysis and evaluation of the proposed method and the features extracted explained in the previous chapter. It is envisaged from the results that the system's performance is largely dependent on the periods in which the data are collected, due to the highest accuracy score being based on a dataset from the last 30 days of the data collection, as investigated in the first experiment. Likewise, the threshold used to extract the sessions based on the flow inter-arrival time, as investigated in the second experiment, as well as the two types of session timing features utilised. The data gathered from the application level for 11 applications show that there is a level of uniqueness in information among users. The system also has the ability to identify users based on their application traffic information achieving the best TPIR, and it was shown that there are promising results whereby the system was able to identify users with an accuracy of 96%. Subsequently, user identification and behaviour profiling based on flow session application timing features is a novel solution to the problem of helping network administrators to monitor and identify users for different security measures and enhance their organisation's policy perspectives.

# **Chapter Six**

## **Overall architecture**

## **6 Overall architecture**

### **6.1 Introduction**

This chapter discusses the architecture of the proposed user behaviour profiling system, which uses application-level flow session timing to identify and profile users by their network traffic (NetFlow). The system components are designed to improve the ability of the system to identify users through network traffic based on the application level, as there is no opportunity to identify users without extracting relevant features. The features extracted need to enhance the way network traffic is dealt with without relying solely on computer network information (IP addresses) directly to identify or classify the individuals who generate the traffic. Another factor that the system needs to deal with is network traffic analysis, as the behaviour and underlying interactions of network applications are constantly changing, with ever more complex patterns being generated. In parallel, user behaviour is also changing and adapting as the online interaction environment changes. As a result, any proposed system to improve user identification and behaviour profiling at the application level should be structured in a framework that connects the user identification and behaviour-profiling components effectively (for more details see section 3.2).

This chapter proposes a theoretical architecture for the system proposed in section 4.2. The chapter also discusses the communication and information required from the engine versus the flow-based communication that flows through the system, from capturing the data to the security aspects the investigator is required to examine. The aim of the proposed framework is to identify users and create a user behaviour traffic profile to support informed decisions made by network security administrators and ISPs regarding policing, traffic management and the various network security perspectives of an organisation-connected network.

## 6.2 The general architecture of user behaviour profiling using an application-level flow sessions

In Chapters Four and Five, two novel session timing features were proposed for user identification behaviour profiling using an application-level flow sessions system. To produce a framework for integrating the two suggested stages in the real world, the following aspects must be defined: data collection engine, traffic pre-processing engine, behaviour classification engine, and security decision engine. The aim of the framework is to integrate the two proposed stages (hour and quarter-hour timing features) in a typical user identification and behaviour profiling system to help investigators identify users and use the user profiling database to make informed decisions from various security perspectives.

The proposed user identification and behaviour profiling system relies on a number of processing engines: a Data Collection Engine, a Traffic Pre-Processing Engine, a Behaviour Classification Engine and a Security Decision Engine, as shown in Figure 6.1.

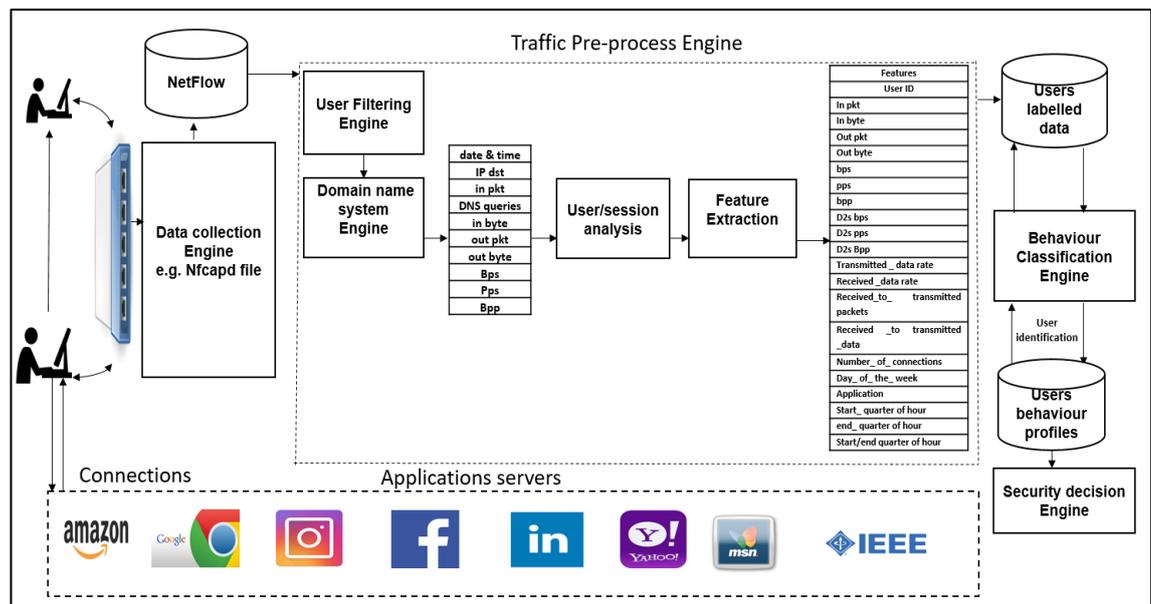


Figure 6.1: Overall architecture for user behaviour profiling using an application-level flow sessions

### 6.2.1 Data Collection Engine

The main duty of the Data Collection Engine is to capture user network traffic metadata (NetFlow) activities. When applications are utilised by a user, the data collection tool, as illustrated in Figure 6.2, automatically gathers the associated information by running the NetFlow collector (nfcapd) on the network switch, including the timestamp, IP source, IP destination, s2d packets, d2s packets, s2d bytes, and d2s bytes. The formats of the NetFlow records are collected and stored in a temporary database to be analysed using the comprehensive process of the proposed method developed by the system. Once the user's traffic data have been collected, the data in temporary storage in nfcapd file format will be removed to be applied in a Linux environment for dumping traffic using the nfdump tool. The temporary storage should have the same basic features as the flow records. Figure 6.2 provides an example of temporary storage with the basin flow records stored in it.

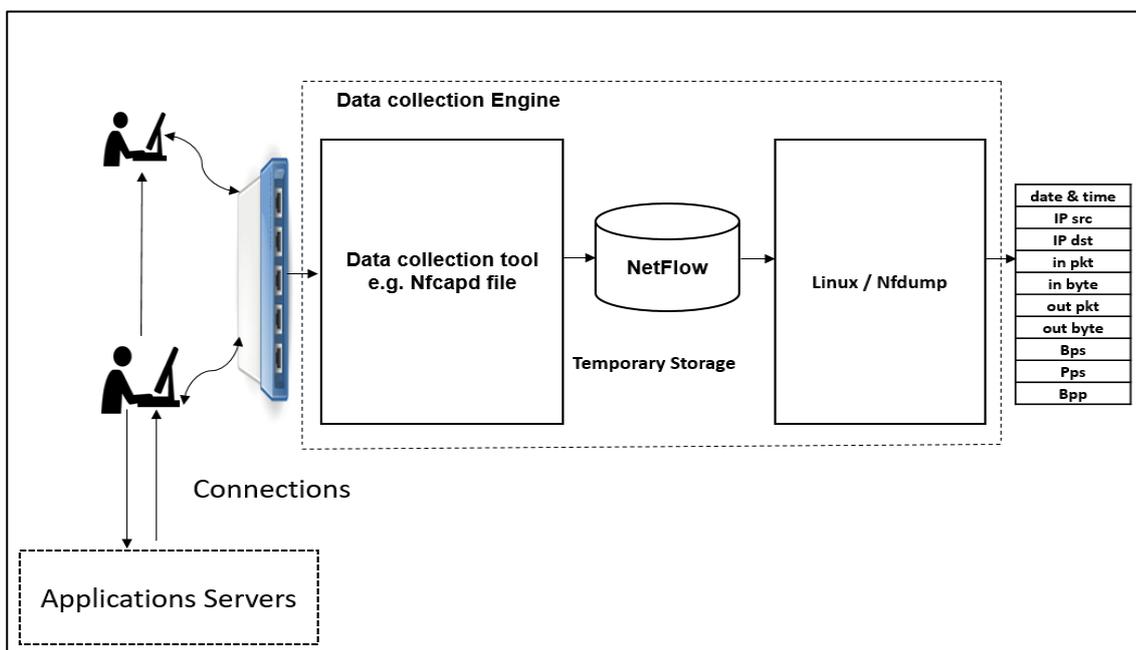


Figure 6.2: Data collection engine

When a user browses the internet, there are thousands of connections starting and ending, as browsing websites (applications) requires connection to the applications' servers. Therefore, the Data Collection Engine collects the traffic transmitted through the router from the user to the server. This flow of data needs to be handled by analysing the user's network traffic. This network traffic metadata are stored on the NetFlow database, as illustrated in Figure 6.2. When the data are stored in the NetFlow database, there is a need to solve the problem of the dynamic changing of source IPs. In order to achieve this, possible solutions were taken into consideration to make sure the source IP does not change during the time to be ready for flowing to the Traffic Pre-Processing Engine.

### 6.2.2 Traffic Pre-Processing Engine

The Traffic Pre-Processing Engine is the 'brain' of the framework, as it contains four enhancement components that help the investigator to handle the user's network traffic until achieving the aim of the proposed framework, as user identification and behaviour profiling is based on application-level flow session timing. The flow of the four components of the Traffic Pre-Processing Engine are illustrated in Figure 6.3.

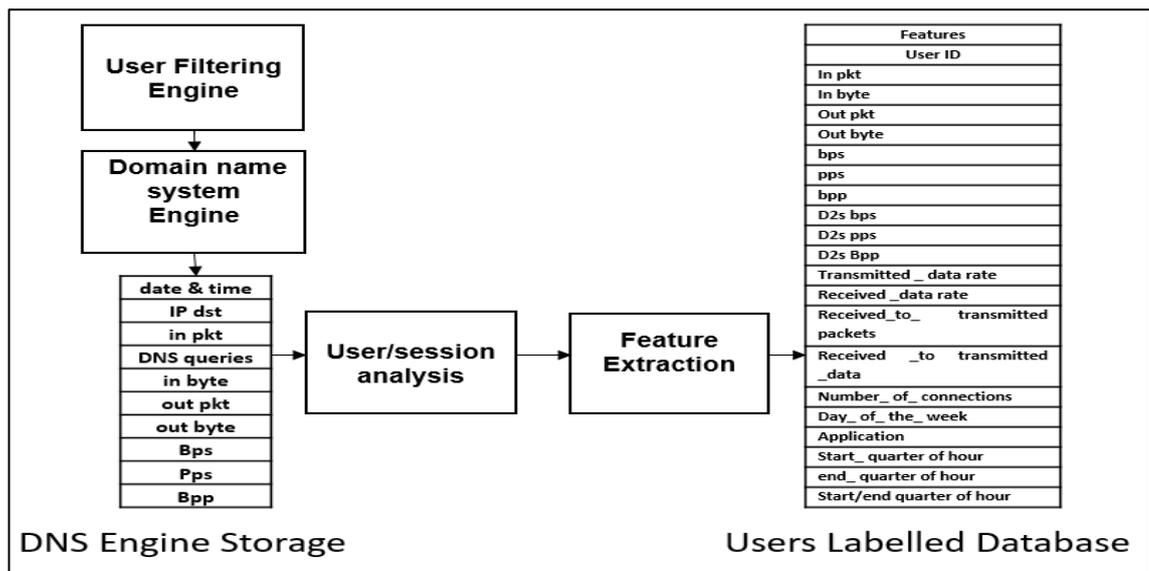


Figure 6.3: Traffic pre-processing engine

As mentioned previously, to confirm the source IPs did not change, the source IP/MAC addresses were mapped as a truth table with timestamps using information derived from an ARP table, by applying the analysis process to the NetFlow database using IP source and MAC mapping criteria (for more details see section 4.4.2). The IP source and MAC mapping process is not applied in any operational version of the proposed system; it is applied here simply to make sure there is no conflict between users. Subsequently, the user-filtering engine, which separates the NetFlow traffic for each user on a daily basis to be ready for the next step, exports the data to the DNS engine, which identifies the applications used by the user. Therefore, the DNS engine is able to enhance the way this problem is dealt with by conducting a DNS lookup for each user’s daily NetFlow traffic as derived from the previous stage (see section 4.4.3). Once the DNS engine has processed and analysed the NetFlow network traffic, the processed data are placed in temporary storage. The DNS engine temporary storage table contains several data fields, such as the date and time, IP destination, in packets, DNS queries, out packets, in bytes, out bytes, bps, pps, and bpp, as shown in Figure 6.3. Table 6.1 illustrates an example of the DNS engine temporary storage with a number of entries.

**Table 6.1: DNS engine storage**

No.	Attribute	Description
1	date & time	Date and start/end time
2	IP dst	IP destination
3	in pkt	Source (client) to the destination (server) number of packets transmitted
4	DNS queries	The DNS query as ‘bbc-vip016.cwwtf.bbc.co.uk.’
5	in byte	Source (client) to destination (server) bytes
6	out pkt	Destination (server) to source (client) number of packets transmitted
7	out byte	Destination (server) to source (client) bytes
9	bps	Source (client) to destination (server) bits per second
10	pps	Source (client) to destination (server) packets per second
11	bpp	Source (client) to destination (server) bytes per packets

The third step in the Traffic Pre-Processing Engine is the user session analysis. The data are forwarded from the DNS engine temporary storage as an input for the user session analysis step. After the DNS engine analysis, the forwarded data are filtered and divided

into web applications (data frames) based on the DNS queries derived from the previous step by filtering the applications used by the user; this step is called application flow filtering and is based on the domain name (for more details see section 4.4.3.1).

The fourth step, which is user/session analysis, is the core of the Traffic Pre-Processing Engine, as the concept of flow session timing is based on the flow inter-arrival time applied to extract the timing feature to identify users based on their 24-hour daily usage using a quarter-hour timing feature. The timing features identify and differentiate users based on web application usage, which varies during the day and in the time spent on each web application.

The results show that the use of quarter-hour timing could be an improvement that would better enable the system to identify users and their application use in different time slots during the day, as the results show in section 5.4. Figure 6.4 shows the user identification of the proposed system for quarter-hour timing features, which scored the highest performance on the system for identifying users based on their daily usage. Once the user has been identified based on the timing features, the user's behaviour profile can be used for various aspects of security, policing, traffic management and traffic monitoring; more details regarding the timing features are discussed in section 3.2.2. The data for the proposed features are stored in the users' labelled data on the database, which then provide input for the Behaviour Classification Engine.

### **6.2.3 Behaviour Classification Engine**

The Behaviour Classification Engine is responsible for exporting the user traffic that was processed and analysed by the previous engines as input for the user behavioural classifier in accordance with the novel timing features extracted and identified on the system and is stored and secured on the users' labelled database. The database contains the data relating to the proposed quarter-hour timing features, which are updated continuously

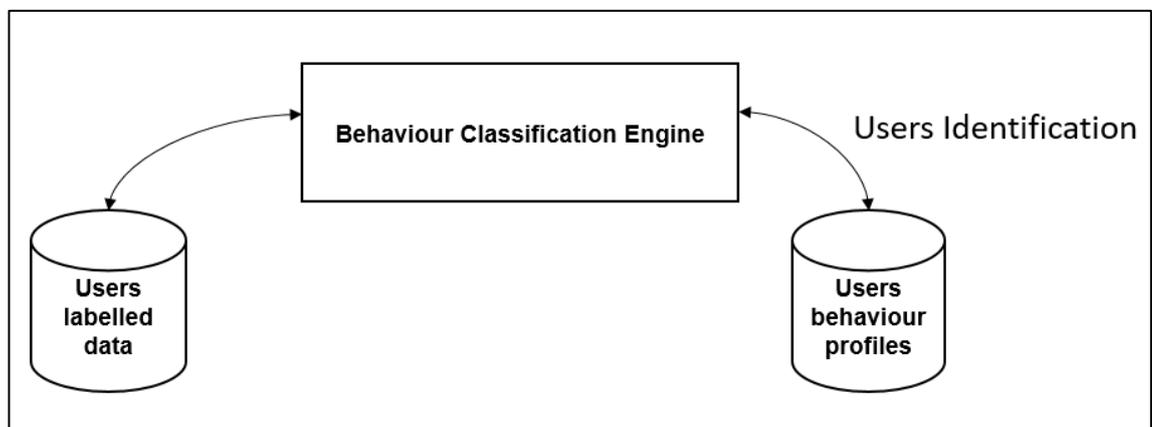
with the users' processed traffic data to be investigated by the Behaviour Classification Engine.

Features
User ID
In pkt
In byte
Out pkt
Out byte
bps
pps
bpp
D2s bps
D2s pps
D2s Bpp
Transmitted _ data rate
Received _ data rate
Received _to_ transmitted packets
Received _to transmitted _data
Number_of_ connections
Day_of_the_ week
Application
Start_ quarter of hour
end_ quarter of hour
Start/end quarter of hour (app usage)

**Figure 6.4: Proposed timing feature**

The Behaviour Classification Engine provides the main functionality for user identification based on the proposed application-level timing features. The user traffic proceeds through all the processes of the Data Collection Engine and the Traffic Pre-Processing Engine until it reaches the users' labelled database, which is continuously updated with newly received traffic. The Behaviour Classification Engine then takes the users' input data (obtained from the users' labelled database), utilising the profiles from the users' behaviour profile database, and is responsible for updating the user profiles after the system has identified a user based on the proposed timing features and web application usage, as shown in Figure 6.5. Users' behaviour profiles are updated each time traffic is received that is related to a specific user based on the records built during the training phase and updates are based on testing the unseen data. The proposed framework

deals with generic network traffic information related to individual users, which includes sensitive information related to an individual. This sensitive information should be stored and used appropriately and the user should have the right to request to have any of the information deleted. The users' behaviour profile database contains sensitive information regarding the users identified on the investigated network, which is protected with a high level of security and a privacy policy to prevent malicious activity. The Behaviour Classification Engine sends the identification results to the Security Decision Engine to make a security response.



**Figure 6.5: Behaviour classification engine**

#### **6.2.4 Security Decision Engine**

The Security Decision Engine is responsible for exporting the user profiles, after which is updated continuously based on the process applied in the Behaviour Classification Engine. The user profiles are stored and managed by the investigator responsible for the security and policy management in the organisation, as updated user profiles are provided by the comprehensive user identification and profiling system. The investigator can be the ISP or the security administrator who manages the Security Decision Engine, as shown in Figure 6.6. The users' behaviour profiles stored on the users' behaviour profile database can help the investigator to make an informed decision regarding policy and

various security perspectives, such as confidentiality, integrity, availability and authorisation.

The user profiling provided by the user identification and profiling system can also help the investigator to enhance the security and performance of the network based on the novel information from the timing features added to the system. Users' profiles will not only contain information about the identity of the user, but might also provide important information about the daily activity of the user. In addition, the time bins that are part of the extracted timing features representing the exact application can be used during the day to gain the precise hours or quarter hours in which activity takes place, which will provide the investigator with a comprehensive view of what the user is doing. Knowing the activity of the user plays an important role in terms of increasing the level of accuracy of the investigated network by helping the investigator to examine both the normal and abnormal behaviour of the users in the organisation.

The Security Decision Engine continuously identifies users through their daily activity as represented in the timing features (quarter hour) and, as demonstrated in the experimental section, provides a high level of performance for the proposed system. The key task of the Security Decision Engine is to monitor current user profiles and make decisions accordingly when an update occurs. The monitoring step conducted by the Security Decision Engine might also help the network administrator to obtain an up-to-date and secure user profile to assist the various network security perspectives applied to the investigated network, as well as managing the organisation's policy.

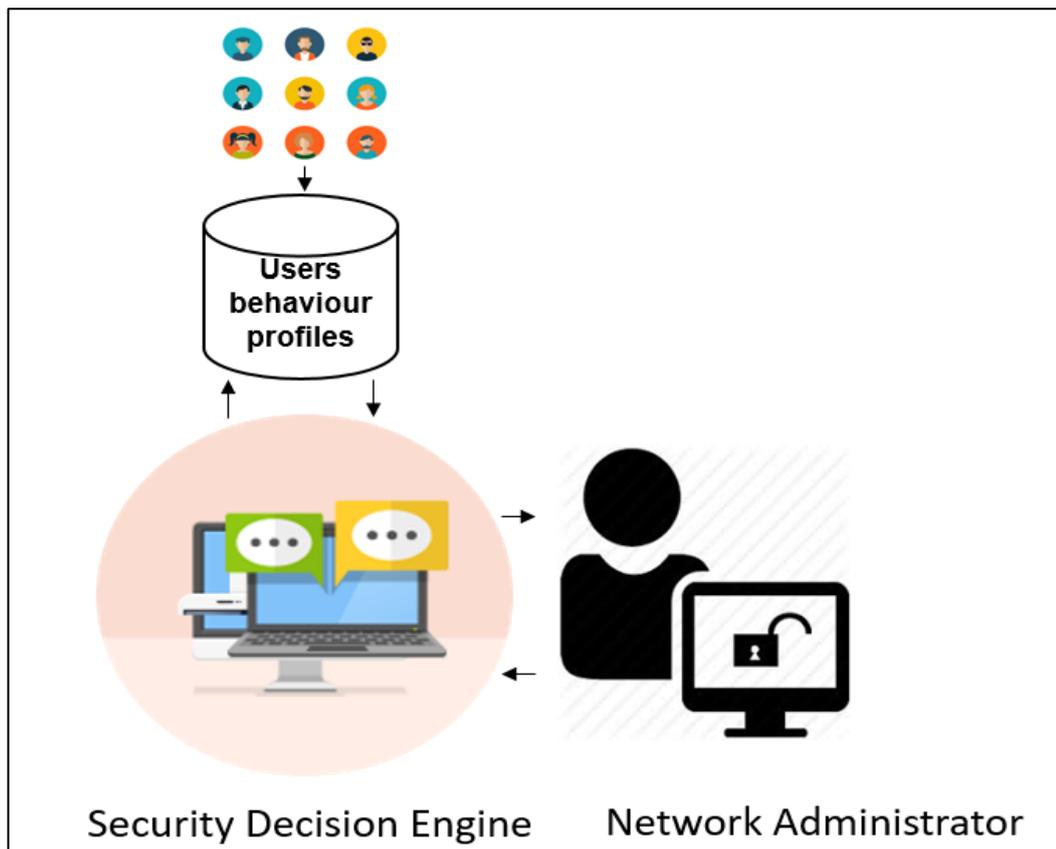


Figure 6.6: Security decision engine

### 6.3 Case study of user behaviour profiling using application-level flow sessions

- Problem:** a company called five - star has several departments. Each department has a manager and 23 employees. Two employees, Mike and John, are using their computer to browse suspicious websites. The network manager needs to know the behaviour pattern of all 23 users by investigating the web applications they visit during the day, as well as the exact time and the amount of traffic exchanged as a profile for each user. However, Mike's and John's behaviour varies and adapts as the online interaction environment changes. The behaviour and underlying interactions of the network applications are constantly changing and user identification and behaviour profiling in real-time network management remain a challenge. The challenge in this case is how to analyse the generic network traffic metadata in order to describe Mike's and John's activity adequately and to what extent their behaviour

could be identified over time. Furthermore, there are complexities around observing the traffic that belongs to a specific user due to traffic noise and dynamic variation in the network resources, such as IPs. Therefore, there is significant interest in finding a way to identify users and generate behaviour profiling from the generic network traffic metadata for traffic engineering and security monitoring. In this case, the need is to investigate the suspicious activity undertaken by Mike and John and compare this with the activities of all the users in the five-star company network investigated.

- **Solution:** to investigate the five-star company network and understand the behaviour activity of the 23 employees in order to understand the suspicious browsing of web applications by Mike and John. For instance, Mike browses Facebook at 9-10 am and 5-8 pm and a suspicious web application at 1-2 pm and 6-7 pm. John browses Facebook at 7-8 am and 10-11 am and a suspicious web application at 11-12 am and 8-9 pm. The proposed system has the ability to investigate the daily usage activity of the employees in the five-star company network by identifying all 23 employees on the network and comparing the behaviour patterns of all the users by focusing on Mike's and John's patterns of behaviour. To do this, the network traffic data (NetFlow) have to be collected by the network administrator for 15 days to investigate the behaviour pattern of the employees on the network, as 15 days are enough to observe changes in users' behaviour. The collected network traffic needs to be separated for each employee in a daily basis to prepare it for the next pre-processing step. The network traffic that has been separated for each employee is then investigated for the web applications that have been used by the users through the DNS queries technique. The web applications used by the users are filtered and classified to be ready for the flow session concept to calculate the features set (hours and quarter-hours), which offers the ability to understand the usage of all users in the company network, as well as Mike and John. Once the whole process is completed, the network administrator

will have a dataset of all 23 employees and their features, including the time bins, which represent the usage activity of each employee based on the extracted statistical features and timing features (Number\_of\_connections, Day\_of\_the\_week, Start\_hour, End\_hour, and 0-23 time bins). The dataset is applied to the Behaviour Classification Engine to predict and train the system to be ready for the unseen data related to each user in order to investigate the suspicious web applications utilised by Mike and John when compared with the other employees in the company. The network administrator will then have a profile of each employee that will act as a basis for any future investigation, which could be updated and employees added, and profiles updated once new traffic is collected.

- **Results:** the network administrator will look at Mike's and John's profiles and understand that Mike browsed a suspicious web application at 1-2 pm and 6-7 pm and John browsed a suspicious web application at 11-12 am and 8-9 pm. In addition, the network administrator will know the exact day that Mike and John browsed the suspicious applications and the amount of traffic exchanged. In general, the combination of information will help the network administrator to examine the employees' profiles and make an informed decision regarding various important network traffic management and policy requirements at the organisation. The administrator could, for example, pass a new policy to the legal department at the company to prevent any behaviour or activity that might compromise or damage the company network.

## 6.4 Conclusion

This chapter presented the overall architecture for user behaviour profiling using application-level flow sessions timing to link the proposed system stages described in Chapter Five. This architecture is composed of a Data Collection Engine, Traffic Pre-Processing Engine (user filtering engine), DNS engine, user/session analysis, features

extraction, Behaviour Classification Engine and Security Decision Engine. These architecture components are integrated through network connections between the applications' servers and the users and managed by the network switch for collection and analysis until achieving novel identification of the system's users and investigating their behaviour profiles. The proposed architecture aims to enhance monitoring and identification of users by using application-level timing features. The identified users and their behaviour profiling can then be investigated by the ISP or security administrator to enable informed decisions about policy, managing the network, and various security perspectives.

The timing features presented in this thesis have shown the ability of the proposed system to identify users, as demonstrated by scoring a high performance, as some of those using web applications in different time slots varied in their behaviour and activity during the day. As a result, extracting a set of features can be seen as the most important point for enhancing any identification and behaviour profiling system, as the strength of such a system depends upon the quality of the extracted features. The following chapter provides a summary of the research project, including its key achievements, limitations, and scope for future work in the user identification and behaviour profiling field.

# **Chapter Seven**

## **Conclusions and future work**

## 7 Conclusions and future work

This chapter concludes the achievements of the research, discusses its limitations and defines future research directions within user identification and behaviour profiling using network traffic.

### 7.1 Achievements of the research

There is increasing interest in identifying users and profiling behaviour from network traffic metadata for traffic engineering and security monitoring. Network security administrators and ISPs need to create user behaviour traffic profiles to make informed decisions about issues such as policing and traffic management and to investigate various network security perspectives. There are, however, serious limitations to this approach. This research has overcome some of the limitations by focusing on flow session timing features to identify and profile users' behaviour from application-level network traffic analysis.

The following points are the main achievements of this research:

- A features extraction process that focuses on identifying and classifying users based on web application-level flow session timing features to represent the users' daily activity. A substantial contribution of this research has been the focus primarily on flow-level-based statistical timing features and not seeking to perform any form of port-based or DPI. The proposed timing features were not only able to identify users' behaviour, but also to represent 24-hour daily user activity based on hour and quarter-hour binary time bins.
- The collection of computer network traffic from real users' interactions to be used in the evaluation phase, as well as the application of DNS query analysis to identify and investigate the web applications used by the users. This was carried out in order to

design and develop a data processing pipeline for identifying network users' profiling based on the features identified.

- An experimental investigation and evaluation of user behavioural profiling using application-level flow sessions by utilising a classification process. This showed that it is possible to identify a good proportion of users successfully given the extracted flow session timing features based on the web application level by utilising a classification process.
- An evaluation and analysis of how identifiable and unique user behaviour profiling is over time in the context of the dynamic and changing nature of network applications. This included investigating the effect of different time resolutions when processing the collected data (hours and quarter hours). The proposed timing feature bins not only identified the user, but also his/her daily 24-hour activities when using an application based on binary encoding representations.
- The creation of a system architecture to guide the system investigator in the connections and several novel processes applied to the system to achieve the aim of this research.
- The proposed system enables a network administrator to profile users' usage patterns by utilising ML algorithms and analysing network traffic. For instance, information such as "Mike tends to look at Facebook at 7-8 am and John looks at Instagram at 5-7 pm" would help a network administrator to examine the two users' profiles and make an informed decision from various network management and security perspectives.

## **7.2 Limitations of the research**

Despite the research objectives having been accomplished, a number of limitations linked to this research can be identified. The following list highlights these limitations:

- The number of users in the experimental dataset was limited. More users and longer profile periods would have provided a more reliable measure of performance than could be achieved.
- While it was not essential for the proposed method to involve an investigation into every web application, having different web application types would have provided a richer and more comprehensive way of investigating users' behaviour profiles.
- An approach that involves investigating users' network traffic raises significant privacy concerns for those users who are monitored by the system. Processing, transmitting and storing samples in a centralised database requires a high level of confidentiality and significant resources.
- The proposed method was analysed and implemented using flow-based network traffic. Another level of analysis that involved using packet-based network traffic might provide another direction for the proposed method.
- The experiments conducted were all performed offline. An evaluation of the proposed system in real-life settings would cover not only the user identification and behaviour activity representations, but also the consideration of other operations, such as the operational overheads of the proposed system with regard to CPU and memory consumption.
- The investigation of users' behaviour patterns needs data to be collected for at least 15 days to train a reliable classifier that is able to distinguish traffic generated by individual users. More investigation is required to understand and detect when the learned user pattern changes. Developing a systematic feature distribution analysis could help in detecting when a user pattern has changed.
- The users in the test environment for this research were mostly (PhD and Master's) students and fell in the 25-45 age group at the University of Plymouth CSCAN lab.

Using another level of analysis with different types of users and age groups, as well as different environments, might provide another direction for the proposed method.

### **7.3 Suggestions and future research scope**

A number of opportunities exist for further research and/or enhancement and these are outlined below:

- The underlying classifiers utilised in the classification stage were based upon a gradient boosting algorithm. Further exploration and investigation of alternative models and algorithms should be conducted to enable improvements in the identification accuracy.
- The number of users in the dataset could be higher to allow the investigation of scalability challenges and to be able to investigate the effect of the method with a different number of users.
- Investigating another type of web application to assess the effect of user identification and behaviour profiling based on the application level with different types of web application would be informative, as these might differ in terms of browsing time and the different network traffic information related to the user profile to complete the profiles and help the investigator from a different perspective.
- Further research could be conducted to investigate the proposed timing feature bins, as well as using different network traffic-level packets to ascertain the effect of packet-level analysis compared with the flow level implemented in this research.
- With the recent advances in deep learning algorithms, it would be interesting to expand the dataset for utilising this type of algorithm to generate more robust and less noisy feature representations for user identification.
- It would be interesting to assess the effects of using a different environment to investigate how the proposed method performs in alternative settings.

## **7.4 The future of research into user identification and behaviour profiling**

Recent years have witnessed a significant interest in identifying users and behaviour profiling from generic network traffic metadata for traffic engineering and security monitoring. Policing, traffic management and various other network security perspectives have increased the interest in identifying users and creating user behaviour traffic profiles to enable network security administrators and ISPs to make informed decisions. Understanding trends in application usage could also be significant in terms of identifying and profiling users in order to represent a particular user's activity through an analysis of network traffic metadata and the extraction of feature sets. However, as behaviour and the underlying interactions of network applications are constantly changing, user identification and behaviour profiling in real-time network management remain a challenge. In parallel, as the online interaction environment changes, this affects user behaviour and activity interactions. Through an in-depth investigation of application-level flow sessions based on DNS filtering criteria and time bins, this research programme finds that behavioural monitoring with an extended set of extracted features can be highly effective in identifying users.

## References

- Afridi, M. W., Ali, Toqeer, Alghamdi, T., Ali, Tamleek and Yasar, M. (2018) 'Android application behavioral analysis through intent monitoring', *6th International Symposium on Digital Forensic and Security, ISDFS 2018 - Proceeding*. IEEE, 2018-Janua, pp. 1–8. doi: 10.1109/ISDFS.2018.8355359.
- Alcock, S. and Nelson, R. (2012) 'Libprotoident: Traffic Classification Using Lightweight Packet Inspection Categories and Subject Descriptors', pp. 1–6.
- Algiriyage, N., Jayasena, S. and Dias, G. (2015) 'Web user profiling using hierarchical clustering with improved similarity measure', in *MERCon 2015 - Moratuwa Engineering Research Conference*. IEEE, pp. 295–300. doi: 10.1109/MERCon.2015.7112362.
- Alotibi, G., Li, F., Clarke, N. and Furnell, S. (2015) 'Behavioral-Based Feature Abstraction from Network Traffic', *Iccws 2015-The Proceedings of the 10th International Conference on Cyber Warfare and Security*, (March 2016), pp. 1–9.
- Alotibi, G., Clarke, N., Li, F. and Furnell, S. (2017) 'User profiling from network traffic via novel application-level interactions', in *2016 11th International Conference for Internet Technology and Secured Transactions, ICITST 2016*, pp. 279–285. doi: 10.1109/ICITST.2016.7856712.
- Auld, T., Moore, A. W. and Gull, S. F. (2007) 'Bayesian neural networks for internet traffic classification.', *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 18(1), pp. 223–239. doi: 10.1109/TNN.2006.883010.
- Bakhshi (2017) 'User-Centric Traffic Engineering in Software Defined Networks', in *2016 23rd International Conference on Telecommunications (ICT)*. 2016 23rd International Conference on Telecommunications (ICT), pp. 1–6. Available at: <http://hdl.handle.net/10026.1/8202>.
- Bakhshi, T. and Ghita, B. (2015) 'User Traffic Profiling: In a Software Defined Networking Context', in *2015 Internet Technologies and Applications (ITA)*. Wrexham, pp. 91–97.
- Bakhshi, T. and Ghita, B. (2016a) 'On Internet Traffic Classification: A Two-Phased Machine Learning Approach', *Journal of Computer Networks and Communications*, 2016. doi: 10.1155/2016/2048302.
- Bakhshi, T. and Ghita, B. (2016b) 'Traffic profiling: Evaluating stability in multi-device user environments', *Proceedings - IEEE 30th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2016*, pp. 731–736. doi: 10.1109/WAINA.2016.8.
- Balam, S. and Wilscy, M. (2014) 'User Traffic Profile for Traffic Reduction and Effective Bot C&C Detection.', *IJ Network Security*, 16(1), pp. 46–52. Available at: <http://isrc.asia.edu.tw:8080/contents/ijns-v16-n1/ijns-2014-v16-n1-p46-52.pdf>.
- Banse, C., Herrmann, D. and Federrath, H. (2012) 'Tracking users on the Internet with behavioral patterns: Evaluation of its practical feasibility', *IFIP Advances in Information and Communication Technology*, 376 AICT, pp. 235–248. doi: 10.1007/978-3-642-30436-1\_20.

- Bermudez, in and Mellia, M. (2012) ‘Dns to the rescue: Discerning content and services in a tangled web’, *Proceedings of the ...*, pp. 413–426. doi: 10.1145/2398776.2398819.
- Bianco, A., Mardente, G., Mellia, M., Munafò, M. and Muscariello, L. (2009) ‘Web user-session inference by means of clustering techniques’, *IEEE/ACM Transactions on Networking*, 17(2), pp. 405–416. doi: 10.1109/TNET.2008.927009.
- Bivens, a, Palagiri, C., Smith, R. and ... (2002) ‘Network-Based Intrusion Detection Using Neural Networks’, *Intelligent ...*, 12, pp. 1–15. Available at: <http://cyberunited.com/wp-content/uploads/2013/03/INTRUSION-DETECTION-USING-NEURAL-NETWORKS.pdf>.
- Bonald, T. (2015) ‘Traffic Models for User-Level Performance Evaluation in Data Networks’, *2015 27th International Teletraffic Congress*, pp. 107–115. doi: 10.1109/ITC.2015.20.
- Bujlow, T., Carela-Español, V. and Barlet-Ros, P. (2015) ‘Independent comparison of popular DPI tools for traffic classification’, *Computer Networks*. Elsevier, 76, pp. 75–89.
- Bujlow, T., Riaz, T. and Pedersen, J. M. (2012) ‘A method for classification of network traffic based on C5.0 machine learning algorithm’, *2012 International Conference on Computing, Networking and Communications, ICNC'12*, pp. 237–241. doi: 10.1109/ICCNC.2012.6167418.
- Carela-Espanol, V., Barlet-Ros, P. and Solé-Pareta, J. (2009) ‘Traffic classification with sampled netflow’, *Peopleacupcedu*, 33(2), p. 34. Available at: [http://people.ac.upc.edu/pbarlet/reports/netflow\\_classification-techrep.pdf](http://people.ac.upc.edu/pbarlet/reports/netflow_classification-techrep.pdf).
- Chappell, L. (2012) *Wireshark ® Network Analysis Wireshark ® Network Analysis*.
- Chi, E. H., Rosien, A. and Heer, J. (2002) ‘LumberJack : Intelligent Discovery and Analysis of Web User Traffic Composition’, *Proceedings of WebKDD*, pp. 1–15.
- Cisco (2016) *Cisco IOS NetFlow - Cisco*. Available at: <http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html> (Accessed: 4 July 2017).
- Cisco (2017) *Solutions - Cisco's 2017 Visual Networking Index (VNI) Infographic - Cisco*. Available at: <http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/vni-infographic.html> (Accessed: 3 July 2017).
- cisco global* (2019). Available at: <https://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html>.
- Cisco VNI Mobile* (2019).
- Cufoglu, A. (2014) ‘User Profiling-A Short Review’, 108(3), pp. 1–9.
- Dainotti, A., Pescape, A. and Claffy, K. C. (2012) ‘Issues and future directions in traffic classification’, *IEEE Network*, 26(1), pp. 35–40. doi: 10.1109/MNET.2012.6135854.
- Dehghani, F., Movahhedinia, N., Khayyambashi, M. R. and Kianian, S. (2010) ‘Real-Time Traffic Classification Based on Statistical and Payload Content Features’, in *2010 2nd International Workshop on Intelligent Systems and Applications*, pp. 1–4. doi:

10.1109/IWISA.2010.5473467.

Deri, L., Martinelli, M., Bujlow, T. and Cardigliano, A. (2014) *NDPI: Open-source high-speed deep packet inspection*, *IWCMC 2014 - 10th International Wireless Communications and Mobile Computing Conference*. doi: 10.1109/IWCMC.2014.6906427.

Deris Stiawan (2012) ‘Intrusion threat detection from insider attack using learning behavior-based’, *International Journal of the Physical Sciences*, 7(4), pp. 624–637. doi: 10.5897/IJPS11.1381.

Du, M., Chen, X. and Tan, J. (2013) ‘Online internet traffic identification algorithm based on multistage classifier’, *China Communications*, 10(2), pp. 89–97. doi: 10.1109/CC.2013.6472861.

Erman, J., Mahanti, A. and Arlitt, M. (2006) ‘Internet Traffic Identification using Machine Learning’, *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, pp. 1–6. doi: 10.1109/GLOCOM.2006.443.

Finsterbusch, M., Richter, C., Rocha, E., Müller, J. A. and Hänßgen, K. (2014) ‘A survey of payload-based traffic classification approaches’, *IEEE Communications Surveys and Tutorials*, 16(2), pp. 1135–1156. doi: 10.1109/SURV.2013.100613.00161.

García-Dorado, J. L., Finamore, A., Mellia, M., Meo, M. and Munafá, M. (2012) ‘Characterization of ISP traffic: Trends, user habits, and access technology impact’, *IEEE Transactions on Network and Service Management*, 9(2), pp. 142–155. doi: 10.1109/TNSM.2012.022412.110184.

Garsva, E., Paulauskas, N., Grazulevicius, G. and Gulbinovic, L. (2014) ‘Packet inter-arrival time distribution in academic computer network’, *Elektronika ir Elektrotechnika*, 20(3), pp. 87–90. doi: 10.5755/j01.eee.20.3.6683.

Gu, X., Yang, M., Fei, J., Ling, Z. and Luo, J. (2015) ‘A Novel Behavior-Based Tracking Attack for User Identification’, in *2015 Third International Conference on Advanced Cloud and Big Data*, pp. 227–233. doi: 10.1109/CBD.2015.44.

Haag, P. (2006) *nfdump and NfSen*. Available at: <http://nfdump.sourceforge.net/>.

Haiyan, Q., Jianfeng, P., Chuan, F. and Rozenblit, J. W. (2007) ‘Behavior analysis-based learning framework for host level intrusion detection’, *Proceedings of the International Symposium and Workshop on Engineering of Computer Based Systems*, pp. 441–447. doi: 10.1109/ECBS.2007.23.

Heer, J., Heer, J., Chi, E. and Chi, E. (2001) ‘Identification of Web User Traffic Composition using Multi-Modal Clustering and Information Scint’, in *Proc. of the Workshop on Web Mining, SIAM Conference on Data Mining*, pp. 51–58.

Heer, J. and Chi, E. H. (no date) ‘Identification of Web User Traffic Composition using Multi-Modal Clustering and Information Scint’.

Hong, Y., Huang, C., Nandy, B. and Seddigh, N. (2015) ‘Iterative-Tuning Support Vector Machine For Network Traffic Classification’, pp. 458–466.

Internet Software Consortium (2004) *host(1) - Linux man page*. Available at: <https://linux.die.net/man/1/host>.

Internet Systems Consortium, I. (2005) *DNS lookup utility UBUNTo, DNS lookup utility*. Available at: <http://manpages.ubuntu.com/manpages/bionic/man1/host.1.html>.

*Internet world stats* (2019). Available at: <https://www.internetworldstats.com/stats.htm>.

Iváncsy, R. and Juhász, S. (2007) 'Analysis of Web User Identification Methods', *World Academy of Science, Engineering and Technology*, 2(3), pp. 338–345.

Jiang, H., Moore, A. W., Ge, Z., Jin, S. and Wang, J. (2007) 'Lightweight Application Classification for Network Management Categories and Subject Descriptors', in *Proceedings of the 2007 SIGCOMM workshop on Internet network management*. INM'07, pp. 299–304.

Jin, Y. *et al.* (2012) 'A Modular Machine Learning System for Flow-Level Traffic Classification in Large Networks', *ACM Transactions on Knowledge Discovery from Data*, 6(1), pp. 1–34. doi: 10.1145/2133360.2133364.

Kabir, S., Mudur, S. P. and Shiri, N. (2012) 'Capturing browsing interests of users into web usage profiles', *AAAI Workshop - Technical Report*, WS-12-09, pp. 18–25.

Kamesh and Sakthi Priya, N. (2014) 'Security enhancement of authenticated RFID generation', *International Journal of Applied Engineering Research*, 9(22), pp. 5968–5974. doi: 10.1002/sec.

Karagiannis, T., Papagiannaki, K. and Faloutsos, M. (2005) 'BLINC: multilevel traffic classification in the dark', *ACM SIGCOMM Computer Communication Review*, 35(4), pp. 229–240. doi: <http://doi.acm.org/10.1145/1080091.1080119>.

Kihl, M. and Odling, P. (2010) 'Traffic analysis and characterization of Internet user behavior', ... *and Control Systems ...*, pp. 224–231. Available at: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5676633](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5676633).

Kirchler, M., Herrmann, D., Lindemann, J. and Kloft, M. (2016) 'Tracked Without a Trace: Linking Sessions of Users by Unsupervised Learning of Patterns in Their DNS Traffic', in *the 2016 ACM Workshop on Artificial Intelligence and Security*, pp. 23–34. doi: 10.1145/2996758.2996770.

Kotsiantis, S. B., Kanellopoulos, D. and Pintelas, P. E. (2006) 'Data preprocessing for supervised learning', *International Journal of Computer Science*, 1(2), pp. 111–117. doi: 10.1080/02331931003692557.

Kounavis, M. E., Kumar, A., Vin, H., Yavatkar, R. and Campbell, A. T. (2004) 'Directions in Packet Classification for Network Processors', *Volume 2 in The Morgan Kaufmann Series in Computer Architecture and Design*, pp. 273–298.

Kumar, G., Kumar, K. and Sachdeva, M. (2010) 'The use of artificial intelligence based techniques for intrusion detection: A review', *Artificial Intelligence Review*, 34(4), pp. 369–387. doi: 10.1007/s10462-010-9179-5.

Kumar, R. (2014) 'Machine Learning based Traffic Classification using Low Level Features and Statistical Analysis', 108(12), pp. 6–13.

Li, B., Springer, J., Bebis, G. and Hadi Gunes, M. (2013) 'A survey of network flow applications', *Journal of Network and Computer Applications*, pp. 567–581. doi: 10.1016/j.jnca.2012.12.020.

- Lim, S. Y. and Jones, A. (2008) 'Network Anomaly Detection System: The State of Art of Network Behaviour Analysis', *2008 International Conference on Convergence and Hybrid Information Technology*, pp. 459–465. doi: 10.1109/ICHIT.2008.249.
- Liu, C. and Wu, J. (2013) 'Fast deep packet inspection with a dual finite automata', *IEEE Transactions on Computers*, 62(2), pp. 310–321. doi: 10.1109/TC.2011.231.
- Lucas, M. W. (Michael W. (2010) *Network flow analysis*. No Starch Press. Available at: [https://books.google.co.uk/books?id=5MDucc0LwiUC&pg=PA35&lpg=PA35&dq=soft+flowd+manual&source=bl&ots=BDQp4yoNFc&sig=BhmOPi5xtFxIKKnfh0R4RItN6rI&hl=en&sa=X&ved=0ahUKEwia\\_sFTg5PUAhWIKsAKHa00BBEQ6AEINDAC#v=onpage&q=softflowd+manual&f=false](https://books.google.co.uk/books?id=5MDucc0LwiUC&pg=PA35&lpg=PA35&dq=soft+flowd+manual&source=bl&ots=BDQp4yoNFc&sig=BhmOPi5xtFxIKKnfh0R4RItN6rI&hl=en&sa=X&ved=0ahUKEwia_sFTg5PUAhWIKsAKHa00BBEQ6AEINDAC#v=onpage&q=softflowd+manual&f=false) (Accessed: 28 May 2017).
- Malott, L. and Chellappan, S. (2014) 'Investigating the fractal nature of individual user netflow data', *Proceedings - International Conference on Computer Communications and Networks, ICCCN*. doi: 10.1109/ICCCN.2014.6911837.
- McDowell, C. M. (2013) *Creating Profiles From User Network Behavior*. Available at: [http://calhoun.nps.edu/bitstream/handle/10945/37673/13Sep\\_McDowell\\_Chad.pdf?sequence=1](http://calhoun.nps.edu/bitstream/handle/10945/37673/13Sep_McDowell_Chad.pdf?sequence=1).
- Medhi, D. (no date) *tcpdump and libpcap latest release*. The Tcpdump Group. Available at: <http://www.tcpdump.org/#latest-release> (Accessed: 10 June 2017).
- Megyesi, P. and Molnr, S. (2012) 'Finding typical internet user behaviors', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7479 LNCS, pp. 321–327. doi: 10.1007/978-3-642-32808-4\_29.
- Melnikov, N. (2010) 'Cybermetrics: User Identification through Network Flow Analysis', *Ifip International Federation For Information Processing*, pp. 167–170.
- Melnikov, N. and Schönwälder, J. (2010) 'User identification based on the analysis of network flow patterns', *Cnds.Eecs.Jacobs-University.De*. Available at: <http://cnds.eecs.jacobs-university.de/courses/nds-2009/melnikov-report.pdf>.
- Mukkamala, S., Janoski, G. and Sung, a. (2002) 'Intrusion detection using neural networks and support vector machines', *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02*, pp. 1702–1707. doi: 10.1109/IJCNN.2002.1007774.
- Nascimento, Z., Sadok, D., Fernandes, S. and Kelner, J. (2014) 'Multi-objective optimization of a hybrid model for network traffic classification by combining machine learning techniques', *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 2116–2122. doi: 10.1109/IJCNN.2014.6889935.
- Nguyen, T. T. T. and Armitage, G. (2008) 'A survey of techniques for internet traffic classification using machine learning', *Communications Surveys & Tutorials, IEEE*, 10(4), pp. 56–76. doi: 10.1109/SURV.2008.080406.
- No, I. and Jamuna, A. (2013) 'Available Online at [www.ijarcs.info](http://www.ijarcs.info) Survey of Traffic Classification using Machine Learning', 4(4), pp. 65–72.
- Oh, S. H. and Lee, W. S. (2003) 'An anomaly intrusion detection method by clustering normal user behavior', *Computers and Security*, 22(7), pp. 596–612. doi:

10.1016/S0167-4048(03)00710-7.

Oliveira, E. de (2011) *Methodologies for Traffic Profiling in Communication Networks*.

Oudah, H., Ghita, B., Bakhshi, T., Alruban, A. and Walker, D. J. (2019) 'Using Burstiness for Network Applications Classification', *Journal of Computer Networks and Communications*, 2019(August), pp. 1–10. doi: 10.1155/2019/5758437.

Oudah, H., Ghita, B. and Bakhshi, T. (2017) 'Network Application Detection Using Traffic Burstiness', in *WorldCIS*.

Paredes-Oliva, I., Castell-Uroz, I., Barlet-Ros, P., Dimitropoulos, X. and Solé-Pareta, J. (2012) 'Practical anomaly detection based on classifying frequent traffic patterns', *Proceedings - IEEE INFOCOM*, pp. 49–54. doi: 10.1109/INFCOMW.2012.6193518.

Park, N. H., Oh, S. H. and Lee, W. S. (2010) 'Anomaly intrusion detection by clustering transactional audit streams in a host computer', *Information Sciences*. Elsevier Inc., 180(12), pp. 2375–2389. doi: 10.1016/j.ins.2010.03.001.

PéterMegyesi, Szabó, G. and Molnár, S. (2015) 'User behavior based traffic emulator: A framework for generating test data for DPI tools', *Computer Networks*, 92, pp. 41–54. doi: 10.1016/j.comnet.2015.09.026.

Phan, M. C., Sun, A. and Tay, Y. (2017) 'Cross-Device User Linking: URL, Session, Visiting Time, and Device-log Embedding', *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 933–936. doi: 10.1145/3077136.3080682.

Piskac, P. and Novotny, J. (2011) 'Network Traffic Classification Based on Time Characteristics Analysis', *Masaryk University Faculty of Informatics*, (January). Available at: [http://is.muni.cz/th/173297/fi\\_r/thesis.pdf](http://is.muni.cz/th/173297/fi_r/thesis.pdf).

Plonka, D. and Barford, P. (2011) 'Flexible Traffic and Host Profiling via DNS Rendezvous', *Workshop SATIN*.

Potdar, K., S., T. and D., C. (2017) 'A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers', *International Journal of Computer Applications*, 175(4), pp. 7–9. doi: 10.5120/ijca2017915495.

Qadeer, M. A., Zahid, M., Iqbal, A. and Siddiqui, M. R. (2010) 'Network Traffic Analysis and Intrusion Detection Using Packet Sniffer', *Communication Software and Networks, 2010. ICCSN '10. Second International Conference on*, pp. 313–317. doi: 10.1109/ICCSN.2010.104.

Qin, T., Guan, X., Wang, C. and Liu, Z. (2014) 'MUCM: Multilevel User Cluster Mining Based on Behavior Profiles for Network Monitoring', *IEEE Systems Journal*, pp. 1–12. doi: 10.1109/JSYST.2014.2350019.

Rossi, D. and Valenti, S. (2010) 'Fine-grained traffic classification with netflow data', *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference on ZZZ - IWCMC '10*, p. 479. doi: 10.1145/1815396.1815507.

Ryan, J., Lin, M. J. and Miikkulainen, R. (1998) 'Intrusion Detection with Neural Networks', *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10*, pp. 943–949.

- Shaikh, Z. A., Science, C. and Science, C. (no date) 'An Overview of Network Traffic Classification Methods'.
- Singh, H. (2015) 'Performance Analysis of Unsupervised Machine Learning Techniques for Network Traffic Classification', *2015 Fifth International Conference on Advanced Computing & Communication Technologies*, pp. 401–404. doi: 10.1109/ACCT.2015.54.
- Sinha, A., Mitchell, K. and Medhi, D. (2003) 'Flow-level upstream traffic behavior in broadband access networks: DSL versus broadband fixed wireless', *Proceedings of the 3rd IEEE Workshop on IP Operations and Management, IPOM 2003*, pp. 135–141. doi: 10.1109/IPOM.2003.1251235.
- Suznjevic, M., Skorin-kapov, L. and Humar, I. (2014) 'User Behavior Detection Based on Statistical Traffic Analysis for Thin client Services', 2, pp. 247–256. doi: 10.1007/978-3-319-05948-8.
- Szabó, G., Turányi, Z., Toka, L., Molnár, S. and Santos, A. (2011) 'Automatic Protocol Signature Generation Framework for Deep Packet Inspection', *Proc. of ICST*, pp. 291–299. doi: 10.4108/icst.valuetools.2011.245606.
- Tao, M., Chun, Y. and Juan, C. (no date) 'Profiling and Identifying Users ' Activities With Network Traffic Analysis', pp. 503–506.
- Therhault, K., Vukelich, D., Farrell, W., Kong, D. and Lowry, J. (2015) 'Network Traffic Analysis Using Behavior-Based Clustering', (November).
- Ulliac, A. and Ghita, B. V. (2010) 'Non-intrusive identification of peer-to-peer traffic', in *3rd Int. Conf. on Communication Theory, Reliability, and Quality of Service, CTRQ 2010, Includes MOPAS 2010: 1st Int. Conf. on Models and Ontology-Based Design of Protocols, Architecture and Services*, pp. 116–121. doi: 10.1109/CTRQ.2010.27.
- Veres, S. and Ionescu, D. (2009) 'Measurement-Based Traffic Characterization for Web 2.0 Applications', in *Scenario*, pp. 5–7.
- Vinupaul, M. V., Bhattacharjee, R., Rajesh, R. and Kumar, G. S. (2017) 'User characterization through network flow analysis', in *Proceedings of the 2016 International Conference on Data Science and Engineering, ICDSE 2016*, pp. 1–6. doi: 10.1109/ICDSE.2016.7823965.
- Wagner, C., State, R., Engel, T. and Learning, M. (2011) 'Machine Learning Approach for IP-Flow Record Anomaly Detection To cite this version : Machine Learning Approach for IP-Flow Record Anomaly Detection', *International Conference on Research in Networking*, pp. 28–39.
- Wang, J. H., An, C. and Yang, J. (2011) 'A study of traffic, user behavior and pricing policies in a large campus network', *Computer Communications*, 34, pp. 1922–1931. doi: 10.1016/j.comcom.2011.05.009.
- Wang, S., State, R., Ourdane, M. and Engel, T. (2011) 'Mining NetFlow Records for Critical Network Activities'.
- Williams, N., Zander, S. and Armitage, G. (2006) 'A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification', *ACM SIGCOMM Computer Communication Review*, 36(5), p. 5. doi:

10.1145/1163593.1163596.

Xu, K., Zhang, Z. and Bhattacharyya, S. (2008) 'Internet Traffic Behavior Profiling for Network', *IEEE/ACM Transactions on Networking*, 16(6), pp. 1241–1252. doi: 10.1109/TNET.2007.911438.

Xue, Y. and Dong, Y. (2013) 'Harvesting unique characteristics in packet sequences for effective application classification', *2013 IEEE Conference on Communications and Network Security (CNS)*, pp. 341–349. doi: 10.1109/CNS.2013.6682724.

Yang, B., Hou, G., Ruan, L., Xue, Y. and Li, J. (2011) 'SMILER: Towards Practical Online Traffic Classification', *2011 ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems*, pp. 178–188. doi: 10.1109/ANCS.2011.34.

Yang, J. *et al.* (2015) 'Characterizing user behavior in mobile internet', *IEEE Transactions on Emerging Topics in Computing*, 3(1), pp. 95–106. doi: 10.1109/TETC.2014.2381512.

Yang, Y. (2010) 'Web user behavioral profiling for user identification', *Decision Support Systems*, 49(3), pp. 261–271. doi: 10.1016/j.dss.2010.03.001.

Zhang, C. X., Zhang, J. S. and Zhang, G. Y. (2008) 'An efficient modified boosting method for solving classification problems', *Journal of Computational and Applied Mathematics*, 214(2), pp. 381–392. doi: 10.1016/j.cam.2007.03.003.

Zhang, F., He, W., Liu, X. and Bridges, P. G. (2011) 'Inferring users' online activities through traffic analysis', *Proceedings of the fourth ACM conference on Wireless network security - WiSec '11*, p. 11. doi: 10.1145/1998412.1998425.

Zhang, H., Lu, G., Qassrawi, M. T., Zhang, Y. and Yu, X. (2012) 'Feature selection for optimizing traffic classification', *Computer Communications*. Elsevier B.V., 35(12), pp. 1457–1471. doi: 10.1016/j.comcom.2012.04.012.

Zhang, J., Xiang, Y., Zhou, W. and Wang, Y. (2013) 'Unsupervised traffic classification using flow statistical properties and IP packet payload', *Journal of Computer and System Sciences*. Elsevier Inc., 79(5), pp. 573–585. doi: 10.1016/j.jcss.2012.11.004.

## Appendix

# Papers

# User Profiling Based on Application-Level Using Network Metadata

Faisal Shaman<sup>1,2</sup>; Bogdan Ghita<sup>1</sup>; Nathan Clarke<sup>1</sup> and Abdulrahman Alruban<sup>1</sup>

<sup>1</sup>Centre for Security, Communications and Network Research  
University of Plymouth, Plymouth, United Kingdom

<sup>2</sup>Faculty of Computers and Information Technology, University of Tabuk, Saudi Arabia  
{faisal.shaman, bogdan.ghita, nathan.clarke, abdulrahman.alruban}@plymouth.ac.uk

**Abstract**— There is an increasing interest to identify users and behaviour profiling from network traffic metadata for traffic engineering and security monitoring. Network security administrators and internet service providers need to create the user behaviour traffic profile to make an informed decision about policing, traffic management, and investigate the different network security perspectives. Additionally, the analysis of network traffic metadata and extraction of feature sets to understand trends in application usage can be significant in terms of identifying and profiling the user by representing the user's activity. However, user identification and behaviour profiling in real-time network management remains a challenge, as the behaviour and underline interaction of network applications are permanently changing. In parallel, user behaviour is also changing and adapting, as the online interaction environment changes. Also, the challenge is how to adequately describe the user activity among generic network traffic in terms of identifying the user and his changing behaviour over time. In this paper, we propose a novel mechanism for user identification and behaviour profiling and analysing individual usage per application. The research considered the application-level flow sessions identified based on Domain Name System filtering criteria and timing resolution bins (24-hour timing bins) leading to an extended set of features. Validation of the module was conducted by collecting NetFlow records for a 60 days from 23 users. A gradient boosting supervised machine learning algorithm was leveraged for modelling user identification based upon the selected features. The proposed method yields an accuracy for identifying a user based on the proposed features up to 74%

**Keywords**- *user profile, user behavioural, user identification, network traffic analysis, supervised learning, network security*

## Introduction

The number of internet users has reached more than five billion across the world, and it is growing continuously [1]. A recent report published by Cisco Inc. in 2016 presented that the traffic data generated was at a level of 7 Exabytes per month, due to reach an expected monthly data volume of 49 Exabyte in 2021 [2]. Due to the massive usage of computer systems and applications, as well as their increased complexity, user identification and

behaviour profiling from generic network traffic have become critical parts of network and traffic management [3]. Primarily for network administrators and security investigators to identify security breaches and enforce the organisation policy as well as provide more intelligent routing decisions for the traffic transiting the infrastructure [4]. User profiling based on the features extracted (source to destination packet size, inter-arrival time) from network traffic metadata encourages the ISP to know the user and how this is reflected in the organisation's security and their policy. User identification and behaviour profiling are the translation of each user activity and include a network footprint of the user interaction. Understanding and identifying subjects from network traffic metadata and profiling their behaviour is a challenging task for researchers as user behaviour while having a common and constant component, also includes slight variations and even the nature of online applications interaction changes over time [5]. In addition, while users can indeed be linked through their authentication profiles with the IP addresses they have allocated, an IP-agnostic solution allows for both a reduction of cross-layer monitoring of users as well as detection of possible intrusion/misuse. This study examines user identification and behaviour profiling by analysing generic network traffic and aiming to profile and identify user behaviour based on their timing and application usage footprint instead of relying exclusively on the IP addressing information [6].

Further in this context, it is worth reminding that both traditional methods of identifying applications, using port-based techniques or Deep Packet Inspection cannot be applied anymore due to the ports randomisation or tunnelling [7] in the case of the former and encryption for the latter [8]. Recent studies [9], [10] focused upon using statistical flow analysis for user identification and behaviour profiling, by extracting features from the flow-level to be able to characterise different users with preservation of user privacy and deal with the encrypted traffic. This relies heavily on the quality of the extracted features and the efficiency of the training phase [11].

The approach proposed in this paper continues the line of research by introducing a novel flow-level set of statistical features set based on the timing of application sessions. The application sessions in turn, are derived from the flow interarrival times and DNS queries. The method aims to improve the accuracy of identifying users and profiling them based on their unique behaviours. The rest of the paper is organised as follows: Section 2 describes the state-of-the-art in traffic classification and user identification and a description with existing limitations of existing approaches. Section 3 explains the proposed method and discusses the rationale for selecting different flow features for this work. Section 4 evaluates the effectiveness of the proposed method by using a supervised machine-learning algorithm with the extracted features. Finally, section 5 concludes the paper and includes possible future work.

### *Related Work*

Historically, the field of user identification and behaviour profiling from generic network traffic information includes a number of different methods and techniques. The first option, a port-based monitoring and profiling is not an option anymore because of the randomly port numbers utilised by different applications are either randomised or tunnelled (towards web-based interfaces), leading to a typical accuracy of less than 70% versus the other available methods (Deep packet inspection and statistical) [12], [13]. It has been argued that the low-accuracy associated with port-based technique can be solved using the Deep Packet Inspection (DPI), which is the most powerful technique on the traffic classification fields as the results showed that the accuracy was very high, up to 95%. However, when dealing with encrypted data, the deep packet inspection techniques can only access the header and metadata of the examined packet [14]. Therefore, this limits the amount of information that such a technique can analyse which in turn affects the identification performance.

The research community has therefore moved towards using statistical methods for instance to overcome the above limitations [9], [15]. A reasonable accuracy of up to 85% was achieved by applying statistical features-based methods such as flow inter-arrival time and packet size to identify users who generated the examined network traffic [16]. The user behaviour profile to be identified from the statistical application levels which have noisy traffic. For instance, when a user concurrently browses multiple websites, his/her behaviour would convolute multiple patterns, increasing the complexity of the user identification task when using the application level [17].

In addition, a number of recent studies [18], [19] have used behavioural profiling in identifying computer network users using DNS information and the volume of traffic, summarised by the number of connections in addition to the statistical overall traffic parameters, by collecting the daily DNS queries and identifying the user sessions with an accuracy of up to 72%. However, the accuracy of user identification based on DNS is also affected by the duration of observation as investigated in [19], which has an accuracy of 73% and 90% with a duration of 65 days and seven days respectively. This is indeed counterintuitive, as the accuracy on the 65 days is lower than the accuracy of seven days. This is potentially due to the slight changes in both user behaviour and application characteristics over time, which jointly may introduce noise on the data. A variety of studies have examined user behaviour profiling from different perspectives such as identification to distinguishing users [18], [20]. The techniques are primarily utilised to identify a user by storing previous user activities to be able to decide whether the examinee user is legitimate. However, the use of the behavioural-based technique by observing the interaction of the client with network applications such as the average packet size while uploading a video on YouTube [21].

Another group of studies have been conducted to explore the possibility of applying user behavioural profiling to increase the level of security in computer networks. Indeed, the early studies in this field have employed an anomaly-based detection to determine any abnormal behaviour [22]. It can be argued that using behavioural profiling can help in differentiating users for various purposes in different performance based on the statistical features extracted from the generic network traffic and the different activities that could be provided to build an accurate user profile[6], [21].

As a result, user behaviour profiling is an appropriate solution in associating with changing of the user behaviour and application over time in a computer network.

To sum up, each method has its strength and limitation based on different circumstances. Relying only in IP addresses or port-based approaches to tag individual may not be useful enough in analysing network traffic.

### **Proposed Method**

This study focuses on extracting and analysing a flow-level feature set that allows identifying user behaviour through its network activity footprint as shown in Figure 1. A set of features is utilised to investigate the users' identification and their daily Internet usage based on a filtered applications session (as explained in subsection B.2). For training and annotating purposes, the used

applications are identified based on DNS queries lookup [23]. The raw network traffic is analysed in terms of representing user's daily usage by using a combination of features based on the session, timing and flow DNS filtering. The concept of user session can be described as a group of continuous flows with characterised by a flow inter-arrival time (i.e. the time between two consecutive flows) lower than a pre-defined threshold.

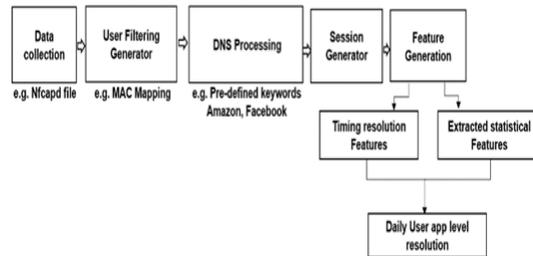


Fig. 1: Proposed User Identification and Behaviour Profiling Methodology

The threshold value is determined by conducting a preliminary analysis that computes the inter-arrival time distribution among the flows. Using session characteristics as a discriminator is based on the fact that user behaviour differs between users (for instance browsing of Facebook varies from user to user in timing and contents). Accordingly, the session is the measure of the variability of user behaviour changing based on timing resolution bins extracted from start/end time of each application sessions (for instance, 24-hour resolution). To validate the method an experiment was carried out using a dataset captured from the University of Plymouth, the Centre for Security Communications and Network Research (CSCAN) lab for 23 users. The raw network traffic was stored as Netflow records using nfdump [24]. The stored flows were pre-processed using Python scripts to filter users based on the MAC/IP address mapping and applications based on DNS queries, and to create additional statistical features. Finally, the dataset was statistically summarised to produce daily user application level records. The newly extracted features were fed into a gradient boosting machine learning algorithm to create a user profile. More details are explained in the next subsections.

### Data Collection

The dataset was collected for 23 users for a period of 60 days (starting from May 8<sup>th</sup>, 2018 till July 8<sup>th</sup>, 2018) based on (ethical approval) approved by a University committee from the student network within the Centre for Security, Communications and Network Research (CSCAN) at Plymouth University, to ensure that the collected data captures most of the user's patterns such as the used applications and variability in their behaviour over time. During this period, the participants accessed

the Internet through the university network and performed their normal daily routine such as browsing and downloading on the Internet. Participants were not asked to follow a protocol, and they merely use their device(s) in their typical fashion. The data was collected during their browsing of the internet and was stored in NetFlow file format, together with the MAC/IP mapping to ensure that IP changes due to DHCP allocation do not affect the accuracy. The top eleven applications were selected based on the statistical procedure, which was computed by implementing the DNS queries keywords for all users to count the connections for each application and choose the top connected and used applications and websites on the lab (i.e., Amazon, Google, Instagram, Facebook, LinkedIn, Yahoo, MSN, Unknown, Stack overflow, TeamViewer, and IEEE). Therefore, these applications were added to the session generator for the applications filtering and labelling purposes. Users were filtered by using the MAC address mapping to label the data related to each use.

### Data pre-processing

The collected data were pre-processed by generating the bidirectional network traffic information. The raw network traffic was generated in several steps regarding getting the most relevant flow-level features to identify the users based on the application sessions and timing resolution criteria. The next subsections explain the undertaken steps to pre-process the raw network traffic to extract desired features.

#### 1) Acquiring Raw Network Traffic

The collected data were initially analysed by *nfdump* tool to generate the daily raw network traffic for all users in the research centre. In addition, the flow records were expanded to get specific bidirectional NetFlow data records including date start/end time, IP source, IP destination, in packets (source to destination packets), in byte (source to destination bytes), out packets (destination to source packets), out byte (destination to source bytes), bps (bits per second), pps (packets per second), and bpp (bytes per packets).

#### 2) Media Accesses Control and IP Source Mapping

DHCP maintained the monitored network for the data collection. Therefore the Netflow collection was accompanied with IP/MAC mapping, to ensure that profiling is allocated to the correct host even if the IP addresses change. Since the MAC address of every hardware is unique, this makes the MAC addresses instead of IPs more reliable to separate the data related to each client for the training

purposes only. Table 1 shows a sample of MAC addresses along with its corresponding IP to keep tracking of the IP assignment. The mapping table is used to ensure that there is no IP conflict occurs through collecting the raw traffic data.

Table 1: MAC Address and IP Source Mapping

Timestamp	Media Access control	IP source
1526029632	b86b23eb1d7f	192.168.200.170

### 3) Domain Name Lookup

The associated domain names are resolved for each Netflow record using a bash script [25]; this is in line with the use of DNS queries were in several previous studies for tracking user behaviour and activity [17], [19]. The DNS lookup utility [23] was utilised on a bash script to initialise the application name (domain name) for each queried flow, by converting IP destination to the domain name. The converted domain name was added as a new attribute (DNS queries) to the Netflow records attributes to be analysed on the next process of this study as shown in Table 2. Therefore, the primary aim of using the DNS lookup utility in this study is to determine which flow belongs to which application that facilitates the automated application flow filtering process.

Table 2: Extracted Features after Domain Name Lookup Process

No.	Attribute	Explanation
1	date & time	Date and start /end time
2	IP dst	IP destination
3	in pkt	source to the destination number of packets transmitted
4	DNS queries	The DNS query as 'bbc-vip016.cwwtf.bbc.co.uk.'
5	in byte	source to destination bytes
6	out pkt	destination to source number of packets transmitted
7	out byte	destination to source bytes
9	Bps	source to destination bits per second
10	Pps	source to destination packets per second
11	Bpp	source to destination bytes per packets

#### a) Application Flow Filtering Based on the Domain Name

The flows were filtered and separated into groups (applications set) based on pre-defined keywords related to the 11 selected popular applications (i.e., Amazon, Google, Instagram, Facebook, LinkedIn, Yahoo, MSN, Unknown, Stack overflow, TeamViewer, IEEE). DNS query results are classified as unknown if the DNS lookup utility generator could not return any value from the given

IP destination address. The applications flow traffic connections that were filtered and combined in data frames (similar to matrix data object) in terms of representing the usage and automate the way of dealing with the raw network traffic for each client's duration as the data related to each user separated in the previous steps. Furthermore, the filtered data frame is used in the session generator for the feature's analysis.

#### b) Session Generator

The filtered applications' data frames are then analysed and divided into sessions using a predefined flows inter-arrival time threshold, assuming that packets in any flow are relatively uniformly spread over the duration of the flow [26]. The flows inter-arrival time is denoted by  $\tau$  (i.e.,  $\tau = \text{the start time of the second flow} - \text{the start time of the first flow}$ ) after converting the date and time to epoch timestamp. The session parameters (Extracted Statistical and Timing Resolution Features section c) were calculated based on a flows inter-arrival time threshold based on two conditions: the flows are part of the same session when the  $\tau$  is less than the threshold (i.e., 10 seconds) and the new session starts when the  $\tau$  is higher than the threshold (i.e., 10 seconds). Furthermore, this procedure is applied in all filtered application data frames in order to divide each application to a set of sessions by generating features based on the session concept

#### A) Features Generation Process

The features set generation, and their discriminative strength is paramount in maximising the accuracy of the user identification. Two types of features, statistical features and session timing-based resolution features were extracted for the dataset. The session timing-based resolution features (hour session start and end) is determined and included within features sets to add another dimension to the features spacing while it could provide the user-dependent pattern.

1) *Extracted Statistical Features:* These features were derived directly from the Netflow records (nfdump) (e.g., source to destination packets and source to destination size of packets as shown in Table 2). Besides, there were other features that were not derived by nfdump, and a calculation was applied to get the complete form of bidirectional flow records data, e.g., destination to source bits per second (bps), destination to source packets per second (pps), destination to source bytes per packets (bpp). Additional features were derived or computed from the above (e.g., Transmitted\_data rate, Received\_to\_transmitted packets). Also, statistical measures were utilised on the extracted statistical flow session features no 1 to 14 (i.e.,

maximum, minimum, mean and median as shown in Table 3).

Table 3: Bidirectional Features Sets

NO.	Features	Explanation
class	User ID	User1, User2, User n
1	In pkt	Session Source to the destination packet
2	In byte	Session Source to destination bytes
3	Out pkt	Session Destination to source packet
4	Out byte	Session Destination to source bytes
5	bps	Session source to destination per second
6	pps	Session source to destination packets per second
7	bpp	Session source to destination per packet
8	D2s bps	Session destination to source per second
9	D2s pps	Session destination to source packets per second
10	D2s Bpp	Session destination to source bytes per packet
11	Transmitted data rate	Session transmitted data rate
12	Received data rate	Session received data rate
13	Received to transmitted packets	Session Received to transmitted packets
14	Received to transmitted data	Session received to transmitted
15	Start time	Session start time
16	End time	Session end time
17	Number_of_connections	Session Number of connections
18	Day_of_the_week	Date encoded from (0-7)
19	Application	Application name encoded (0-10)
20	Start_hour	Integer encoded from (0-23)
21	end_hour	Integer encoded from (0-23)
22	Start/end hour	Start / end hour integer from (0-23) represented on (0-1) timing bins

2) *Session Timing Resolution Features*: The timing based features were extracted based on the start and end time of the sessions that are proposed by this study, and it includes two types of features relating to the user activity characteristics: session activity and application usage features number 15 to 22 as shown in Table 3.

a) *Daily User Session Encoding*: Once the session was generated as defined on section 2 and features sets (Extracted Statistical Features, Timing Resolution Features) were extracted for each application's session based on the process explained on the previous sections. Then, the start/end time (hour) was extracted into a separated Feature as an integer that represents the hour (0-23), as shown in Figure 2. The 24-hour was encoded in terms of combining the start and end timing resolution for the whole sessions related to one application, to represent the daily usage as explained in the next section. Furthermore, the feature that represents applications was encoded into an integer based on the initialised application name to be able to operate with many machines learning which require input as numeric rather than labels, by converting each categorical value into numerical (0-10). Also, the day feature is encoded

from (0-7) to represent the day of the week. The data can be described as nominal features, e.g., applications name or numerical, e.g., 0, 1 and 2. While some classification algorithms can work with nominal features, such as the Decision Tree or the Random Forest, almost all can work on numerical ones, such as the Support Vector Machine or the Multilayer Perceptron. This makes it necessary to encode the nominal to numerical features.

User	Extracted Stastical Features				24-hours Timing Encoding			
	(Max, Min, Mean, Median)				App	Day	Start_hour	End_hour
1	.....	.....	.....	.....	0	0	0	0
1	.....	.....	.....	.....	0	0	1	1
1	.....	.....	.....	.....	1	0	.	.
2	.....	.....	.....	.....	2	1	.	.
2	.....	.....	.....	.....	4	1	23	23

Fig.1: Daily user session resolution

b) *Daily User app level resolution*: The daily user application level time resolution features were encoded into (0, 1) timing bins as shown in Figure 2. To combine all sessions related to the applications filtered and pre-processed by representing the user's daily usage behaviour. Also, to gain a higher user's daily application's activity resolution, the mean of each application's session extracted statistical features was calculated. This allowed summarising the activity of a user for one day in a single record. In addition, in this stage, the start/end hour was converted to binary encoding to represent the daily app level time resolution. For instance, if Amazon (0) used from an hour (0-9) and (20-23) per day, all these hour bins will be given 1s, and other bins will take 0s. Furthermore, if Facebook (1) is used based on an hour (10-15), this hour bins will take 1's, and other bins will take 0's.

User	Extracted Stastical Features				24-hours Timing Resulation Features							
	(Max, Min, Mean, Median)				App	Day	Start hour	End hour	hour (0-9)	hour (10-15)	hour (16-19)	hour (20-23)
1	.....	.....	.....	.....	0	0	0	9	1	0	0	1
1	.....	.....	.....	.....	1	0	10	15	0	1	0	0
1	.....	.....	.....	.....	2	0	16	19	1	0	1	0
1	.....	.....	.....	.....	3	0	20	23	0	0	0	1

Fig.2: Daily User App Level Resolution

## Evaluation

Gradient boosting is a useful practical supervised machine learning for different predictive tasks, and it can dependably provide more accurate results than the straight single machine learning models which are inspired by the gradient boosting framework of [27], which has been previously applied to solve classification and regression problems and more recently to train conditional random fields. The boosting supervised machine learning was utilised to build a series of small decision trees based on the collected data and each tree attempts to correct errors from the previous stage. During the last few years, many practical

studies were published, which use decision trees as the base learning for gradient boosting [26], [27]. Furthermore, the algorithm can optimise any differentiable loss function by using a gradient descent approach [28]. This approach builds the trees sequentially to sum an individual tree consecutively, which provide the best solution under different conditions. In addition, the Z-score was applied to the dataset to normalise the numeric data, excluding the binary bins features for higher accuracy on the end classification model [29]. The data were split randomly into two sets; 70 % of the data were used to train the gradient boosting classifier while 30% of the data were used for testing between all user's data. The classifier performance was evaluated with different metrics derived from the four parameters: True positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). The evaluation parameters (accuracy, precision, recall, and F1 score) were calculated based on error rates, which are represented in the confusion matrix according to the following equation:

- Accuracy: it is the one that predicts the overall accuracy of the model.

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

- Precision: it gives the fraction of the classifier prediction that is true.

$$(2) \quad Precision = \frac{TP}{TP+FP}$$

- Recall: The percentage of true results out of all results estimated by the classifier.

$$(3) \quad Recall = \frac{TP}{TP+FN}$$

- F1 score: it is a metric conducted by calculation of the Precision and Recall and more useful than accuracy in case of uneven multiclass distribution and if the FP and FN are very different [9].

$$(4) \quad F1 \text{ score} = \frac{Recall * Precision}{Recall + Precision}$$

#### A. Experimental Results:

The results as shown in Table 4 that the flows generic network traffic analysis can produce a notable result in terms of user identification and behaviour profiling. In comparison with previous studies [9], [17], in [9] the achieved accuracy was 73% by using flow network analysis approach, while our study achieved up to 74% level of accuracy. Therefore, different aspects affected the accuracy between our study and compared studies for instance (volume of traffic, the environment of

collecting the flow network traffic). Table 4 shows the accuracy of all feature's sets to up to 74%. The accuracy of set 2 which represents the second 30 days of the data exceeded the accuracy of set 1 and set 3 which represent the first 30 days and all the 60 days of the data. Therefore, set 1 and set 3 data were affected by less traffic generated by the experimental lab users (due to a holiday) while set 2 traffic generated by users were normal, and these affect the volume of interactions of the examined users in those sets. The highest accuracy on set 2 improves the proposed measurement features that were affected by the periods of collecting data and user access limitation, which was observed on the volume of traffic data between all sets.

Table 4: Users' Traffic Classification Results

Set	No. users	No. days	Accuracy	Precision	Recall	F 1 score
Set 1	23	1 <sup>st</sup> /30	68%	64%	63%	63%
Set 2	23	2 <sup>nd</sup> /30	74%	75%	73%	73%
Set 3	23	60	72%	67%	65%	65%

The classification comparison was implemented by the gradient boosting using the set 2 features as shown in Table 5. The comparison indicates the extracted statistical features and session timing resolution features by employing them to the classifier separately. The session timing resolution features indicated the highest usage score of up to 65% compared to the extracted statistical features which were up to 61%. The session timing resolution attributes were scored the highest usage among all users. Also, the set 2 features were applied to random forest feature importance, which indicated a good performance between all features to identify users.

Table 5: Classification Performance for Each Feature type

Feature type	Accuracy	Precision	Recall	F1 score
Timing resolution features	65%	62%	60%	60%
Extracted statistical features	61%	59%	53%	55%
Both	74%	75%	73%	73%

Therefore, the top 10 features are represented in Figure 4, in which the first top 4 features (app\_encoded, end\_hour, start\_hour, number\_of\_connections) were scored the highest usage between extracted statistical and proposed session timing resolution features based on the whole dataset. The features importance analysis applied to the proposed timing resolution features indicated by the highest usage of the features was because that the proposed timing features enhanced the classifier to identify and discriminate users. Therefore, the highest score achieved with the two-feature type (Timing resolution features and extracted statistical features), which indicated that the module was being enhanced by the proposed features to differentiate between user's traffic

samples. The `app_` encoded feature enhanced the module to identify users who indicate that the encoding criteria applied on the features on the feature extracting step. Also, the `start_hour` and `end_hour` features scored the second and third top highest between all features which indicate the importance of the 24-hour timing resolution features to identify users from their investigated traffic samples.

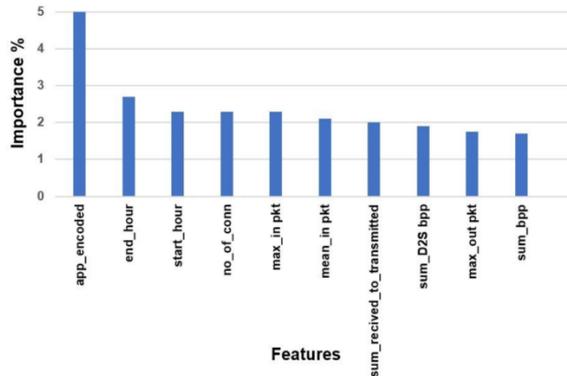


Fig.3: Random Forest Feature Importance

the classification model is ideally high between all classes ranging from 48 -100%. The labels are indicated with users' id from (User 1- User 23) as illustrated in Table 6 for the predicted and true labels. The highest score of the actual class predict 97% as the TP for User 10, and 3% of FN recorded for users (4 and 12), which indicate the ability of the module to identify users with a high score.

Furthermore, User 6 scored the second highest accuracy between all users which is 91% of TP classified samples, there were 9% FN misclassified recorded for User 23. Also, User 3 was recorded as the third highest score on the module with 90% TP and 14% misclassified attributed to users (2 and 20). User 23 recorded the lowest accuracy with 48% TP and 52% FN misclassified attributes to users (2, 4,7,13 and 18) on the module because of the number of traffic samples are the lowest between all users on the module. The reason of the small number of User 23 traffic samples are the number of days which affect the TP for this user comparing to others as there is no traffic for the whole 60 days depends on his usage which is lower than other users on the module.

### B. Confusion Matrix

The most straightforward way to evaluate the performance of the classifier is based on a confusion matrix especially when the model has more than two classes. Table 6 illustrates a confusion matrix for all users to show the correct and incorrect prediction of each class based on the test data for set 2 features set. The performance of

Table 6: Confusion Matrix (Features Set 2)

1	67	11	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0
2	0	85	0	0	5	0	0	0	0	0	0	5	0	0	0	0	0	0	0	5	0	0
3	0	6	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
4	0	0	0	70	0	0	0	0	0	0	20	0	10	0	0	0	0	0	0	0	0	0
5	0	0	0	5	58	0	0	0	6	0	0	0	0	0	5	21	0	0	0	5	0	0
6	0	0	0	0	0	91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9
7	0	0	0	0	0	0	75	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0
8	5	0	0	0	5	0	0	72	0	0	0	0	3	0	0	0	0	0	10	0	0	0
9	0	0	0	0	0	0	7	0	75	0	0	0	0	0	0	0	0	0	7	0	0	4

TRUE LABEL

10	0	0	0	2	0	0	0	0	0	97	0	1	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	3	0	0	0	0	0	0	87	0	0	0	0	0	7	0	0	0	3	0	0	0
12	0	0	0	11	0	0	0	0	0	0	0	89	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	7	0	0	7	0	0	57	0	0	8	0	0	0	21	0	0	0	0
14	0	0	0	14	0	0	8	0	0	0	7	0	0	71	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	15	0	0	71	0	0	14	0	0	0	0	0	0
16	0	0	0	11	0	0	6	0	6	0	0	0	6	0	0	68	0	0	3	0	0	0	0	0
17	0	0	0	4	4	0	4	0	0	0	6	0	0	4	0	0	78	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	88	2	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	18	0	65	0	0	0	8	0	0
20	0	0	0	8	0	0	0	0	8	0	0	0	0	0	7	0	0	0	0	71	0	0	0	6
21	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	10	0	0	60	0	0	0	0
22	0	0	15	0	0	0	0	10	0	0	0	2	0	0	0	0	15	0	0	0	58	0	0	0
23	0	12	0	15	0	0	10	0	0	0	0	0	10	0	0	0	0	5	0	0	0	0	0	48
Uid	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	23

Predicted Label

## V. Discussion

The experiment results indicated that the nature of the features derived from flow-level generic network traffic is unique, thereby using it to identify and build a user behavioural profile is a promising solution to help the security administrator to make an informed decision about different perspectives. In addition, the proposed features and the analysis of the user traffic information can enhance user identification and behaviour profiling. Moreover, the experiment showed that by utilising session timing resolution and extracted statistical features, the system was able to identify users and represent the usage of applications to up to 74 % level of accuracy as shown in Table 4. Since set 2 achieved a high level of accuracy, it is a clear indication that there is discriminative information exists in the user proposed features predicted as a right classification module. The confusion matrix description of the high attributed level of the truly predicted classes indicates the level of unique information among different users and application usage. Therefore, obtaining a precise user statistical and session timing resolution pattern lead to an accurate module that helps to identify profile users.

Also, observation and calculation of the user behaviour activity among different applications provide a robust approach to identify relevant traffic. When combined with the DNS queries, user MAC address mapping and session timing resolution (an approach uses the DNS queries to initialise applications and then identify the user data that uses session information to tag all traffic from that user) provides a very successful approach to the target upon the traffic that is most relevant.

The analysis relied on MAC addresses instead of IP addresses to ensure host consistency, as the

(DHCP) changes IP address among users during the time. The system successfully identified the individual user with accuracies of 48 - 100% as demonstrated in Table 6. Therefore, the accuracy differs among users due to the volume of traffic and the period of collecting the data, as some of the users scored higher accuracy compared to other users. The explanation of predicted label being high was attributed to the level of uniqueness of users in an application. The applications analysis of this approach was identified based on DNS queries which were implemented relying on DNS lookup utility, which is a good objective in case millions of records needs to be investigated. Moreover, an automated way of dealing with the real traffic used in this approach and provide the ability to deal with any number of users on the investigated network.

## VI. Conclusion and Future Work

The present work proposes a method for user identification and behaviour profiling from generic network traffic. The resulted classification accuracy shows that the proposed features based on application-level flow sessions could be utilised to discriminate among users with an accuracy of up to 74%. A supervised machine-learning algorithm was employed to evaluate the analysis algorithm with real data collected from the Centre for Security, Communications and Network Research (CSCAN) at Plymouth University to investigate the proposed approach.

Apart from the future work, implement different timing resolutions such as (quarter\_of\_hour) features to see the effect of the new features and different distribution analysis will be applied on the sessions flow inter-arrival time, to investigate the impact of different thresholds and its effect on system performance. Additionally, more experimental work and analysis will be utilised to examine the effect of each users features based on variance and similarity based on the natures of features.

## Reference

- [1] J. Yang, Y. Qiao, X. Zhang, H. He, F. Liu, and G. Cheng, "Characterizing user behavior in mobile internet," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 1, 2015, pp. 95–106.
- [2] Cisco, "Solutions - Cisco's 2017 Visual Networking Index (VNI) Infographic - Cisco," 2017. [Online]. Available: <http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni-infographic.html>. [Accessed: 03-Jul-2017].
- [3] T. Bakhshi and B. Ghita, "User Traffic Profiling: In a Software Defined Networking Context," in *2015 Internet Technologies and Applications (ITA)*, 2015, pp. 91–97.
- [4] T Bakhshi, B Ghita, User-centric traffic optimization in residential software defined networks, *2016 23rd International Conference on Telecommunications (ICT)*, (2016), pp. 1-6.
- [5] T Bakhshi, B Ghita, Traffic Profiling: Evaluating Stability in Multi-device User Environments, *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)* (2016), pp. 731-736.
- [6] G. Alotibi, N. Clarke, F. Li, and S. Furnell, "User profiling from network traffic via novel application-level interactions," in *2016 11th International Conference for Internet Technology and Secured Transactions, ICITST (2016)*, pp. 279-285 .
- [7] F. Dehghani, N. Movahhedinia, M. R. Khayyambashi, and S. Kianian, "Real-Time Traffic Classification Based on Statistical and Payload Content Features," in *2010 2nd International Workshop on Intelligent Systems and Applications*, 2010, pp. 1–4.
- [8] M. Finsterbusch, C. Richter, E. Rocha, J. A. Müller, and K. Hänßgen, "A survey of payload-based traffic classification approaches," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 2, 2014, pp. 1135–1156.
- [9] M. V. Vinupaul, R. Bhattacharjee, R. Rajesh, and G. S. Kumar, "User characterization through network flow analysis," in *Proceedings of the 2016 International Conference on Data Science and Engineering, ICDSE 2016*, pp. 1-6 .
- [10] A. Ulliac and B. V. Ghita, "Non-intrusive Identification of Peer-to-Peer Traffic," 2010 Third International Conference on Communication Theory, Reliability, and Quality of Service, Athens/Glyfada, 2010, pp. 116-121
- [11] T Bakhshi, B Ghita, "On internet traffic classification: A two-phased machine learning approach", *Journal of Computer Networks and Communications*, 2016
- [12] J. Erman, A. Mahanti, and M. Arlitt, "Internet Traffic Identification using Machine Learning," *Global Communications Conference 2006. GLOBECOM '06. IEEE*, 2006, pp. 1–6.
- [13] M. E. Kounavis, A. Kumar, H. Vin, R. Yavatkar, and A. T. Campbell, "Directions in Packet Classification for Network Processors," *Vol. 2 Morgan Kaufmann Series in Computer Architecture and Design*, 2004, pp. 273–298.
- [14] PéterMegyesi, G. Szabó, and S. Molnár, "User behavior based traffic emulator: A framework for generating test data for DPI tools," *Computer Networks*, vol. 92, 2015, pp. 41–54.
- [15] N. Melnikov, "Cybermetrics: User Identification through Network Flow Analysis," *Ifip International Federation For Information Processing*, 2010, pp. 167–170.
- [16] H Oudah, B Ghita, "Network Application Detection Using Traffic Burstiness", *World Congress on Internet Security WorldCIS-2017* (2017), pp. 23 – 28.
- [17] C. M. McDowell, "Creating Profiles From User Network Behavior," 2013.
- [18] C. Banse, D. Herrmann, and H. Federrath, "Tracking users on the Internet with behavioral patterns: Evaluation of its practical feasibility," *IFIP Advances in Information and Communication Technology*, vol. 376 AICT, 2012, pp. 235–248.
- [19] M. Kirchler, D. Herrmann, J. Lindemann, and M. Kloft, "Tracked Without a Trace: Linking Sessions of Users by Unsupervised Learning of Patterns in Their DNS Traffic," in *the 2016 ACM Workshop on Artificial Intelligence and Security*, 2016, pp. 23–34.
- [20] X. Gu, M. Yang, J. Fei, Z. Ling, and J. Luo, "A Novel Behavior-Based Tracking Attack for User Identification," in *2015 Third International Conference on Advanced Cloud and Big Data*, 2015, pp. 227–233.
- [21] G. Alotibi, F. Li, N. Clarke, and S. Furnell, "Behavioral-Based Feature Abstraction from Network Traffic," *Iccws 2015-The Proceedings of the 10th International Conference on Cyber Warfare and Security 2015*, pp.1-9.
- [22] S. H. Oh and W. S. Lee, "An anomaly intrusion detection method by clustering normal user behavior," *Computers and Security*, vol. 22, no. 7, 2003, pp. 596–612.
- [23] D. lookup Utility, "DNS lookup utility UBUNTo," *DNS lookup utility*, 2005. [Online]. Available: <http://manpages.ubuntu.com/manpages/bionic/man1/host.1.html>.
- [24] P. Haag, "nfdump and NfSen," 2006. [Online]. Available: <http://nfdump.sourceforge.net/>.
- [25] Internet Software Consortium, "host(1) - Linux man page." [Online]. Available: <https://linux.die.net/man/1/host>.
- [26] E. Garsva, N. Paulauskas, G. Grazulevicius, and L. Gulbinovic, "Packet inter-arrival time distribution in academic computer network," *Elektronika ir Elektrotechnika*, vol. 20, no. 3, 2014, pp. 87–90.
- [27] C. X. Zhang, J. S. Zhang, and G. Y. Zhang, "An efficient modified boosting method for solving classification problems," *Journal of Computational and Applied Mathematics*, vol. 214, no. 2, 2008, pp. 381–392.
- [28] Y. Yang, "Web user behavioral profiling for user identification," *Decision Support Systems*, vol. 49, no. 3, 2010, pp. 261–271.
- [29] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data preprocessing for supervised learning," *International Journal of Computer Science.*, vol. 1, no. 2, 2006, pp. 111–117.

# User Profiling Using Application-Level Sessions Timing Resolution

Faisal Shaman<sup>1,2</sup>; Bogdan Ghita<sup>1</sup>; Nathan Clarke<sup>1</sup> and Abdulrahman Alruban<sup>1</sup>

<sup>1</sup>Centre for Security, Communications and Network Research  
University of Plymouth, Plymouth, United Kingdom

<sup>2</sup> Faculty of Computers and Information Technology, University of Tabuk, Kingdom of Saudi Arabia  
{faisal.shaman, bogdan.ghita, nathan.clarke, abdulrahman.alruban}@plymouth.ac.uk

**Abstract**— User identification and behaviour profiling from generic network traffic is a critical step that allows the ISP or security administrator, to take into consideration the information and make an informed decision about policing, traffic management, and enforcing the policy of the organisation. Additionally, application usage trend is significant in terms of identifying and profiling the user by analysing the generic network traffic and extracting a relevant feature that represents the user's activity. However, user identification and behaviour profiling in real-time network management remains a challenge, as the behaviour and underline interaction of network applications are permanently changing. In parallel, user behaviour is also changing and adapting, as the online interaction environment changes. Also, the challenge is how to fully describe the user activity among generic network traffic in terms of identifying the user and his changing behaviour over time. In this paper, we propose a novel mechanism for user identification and behaviour profiling from generic network traffic. The research considered the application-level flow sessions identified based on Domain Name System (DNS) filtering criteria and a timing resolution bins leading to an extended set of features. Validation of the module was conducted by collecting NetFlow records over a 60-day period from nine users. The Gradient Boosting supervised machine learning classifier was utilised to train and test the selected features. The average results of identifying a user based on the proposed features between all ranks range from 67-91%.

**Keywords**- user profile, user behavioural, user identification, network traffic, network security

## Introduction

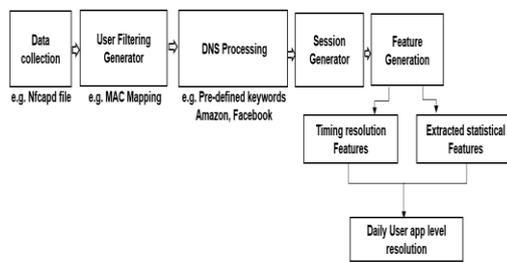
The number of internet users has reached more than five billion across the world, and it is growing continuously (Yang *et al.*, 2015). A recent report published by Cisco Inc. in 2016 presented that the generated data traffic was at a level of 7 Exabyte per month. Subsequently, the growth rate increased to 11 Exabyte per month in 2017, with an expected monthly data growth of 49 Exabyte in 2021(Cisco,

2017). Due to the massive usage of computer systems and applications, as well as their increased complexity, user identification and behaviour profiling(Bakhshi and Ghita, 2015) from generic network traffic have become critical for network managers and ISPs to identify security breaches and enforce the organisation policy as well as provide more intelligent routing decisions for the traffic transiting the infrastructure (Bakhshi, 2017). User identification and behaviour profile is the translation of each user's activity, and it includes a diversity of user information such as personal, interests, and preference data. Understanding and identifying users from generic network traffic and profiling their behaviour is a challenging task for researchers as the user's behaviour is not permanently constant, even the nature of online applications interaction changes(Bakhshi and Ghita, 2016b). Therefore, this study looks for user identification and behaviour profiling by analysing generic network traffic, though relying only on the Internet Protocol (IP) addresses to directly identify users who generated such traffic is not reliable due to the dynamic change of IP's (Alotibi *et al.*, 2017).

In addition, the port-based technique which is the oldest and the most common method for identifying traffic based on analysing the used communication ports assigned by Internet Assigned Numbers Authority (IANA) cannot be used due to the ports randomisation (Dehghani *et al.*, 2010). In addition, the Deep Packet Inspection approach is one of the most important techniques used for identifying the user traffic based on application level analysis, as it could go deeply on the content of packet, but it needs specific resources and cannot be allocated with the encrypted traffic (Finsterbusch *et al.*, 2014). Recent studies [8 -10] were focused upon using statistical flow analysis method for user identification and behaviour profiling, by extracting features from the flow-level in order to be able to characterise different users with preservation of user privacy and deal with the encrypted traffic. According to the statistical flow

analysis method by using the Machine learning algorithm to identify users, the quality of the extracted features and training phase makes a significant effect on the predicted module performance(Bakhshi and Ghita, 2016a).

The proposed system introduces novel flow-level statistical features based on the application sessions, DNS queries and timing resolution for improving the user identification and behaviour profiling. The rest of the paper is organised as follow. Section 2 describes the state-of-art in traffic classification and user identification to describe the limitations of existing approaches. Section 3 explains the proposed approach and discusses the rationale for selecting different flow features for this work. Section 4 evaluates the developed machine-learning model by using a supervised machine-learning algorithm with the extracted features. Finally, section 5 concludes and represents the future work.



Proposed User Identification and Behaviour Profiling Methodology

### Related Work

Previous studies in the field of user identification and behaviour profiling from generic network traffic information area proposed numerous methods and techniques to combat and reduce the user traffic identification issues. The port-based technique is not sufficient nowadays because of the randomly port number utilised on the different application as port-based is considered valid with the highest accuracy of 75% compared with other available methods [10, 11]. It has been argued that the low-accuracy associated with port-based technique can be solved using the Deep Packet Inspection (DPI), which is the most powerful technique on the traffic classification fields as the results showed that the accuracy was very high, up to 95%. However, when dealing with an encrypted data, the deep packet inspection techniques can only access the header and metadata of the examined packet (PéterMegyesi, Szabó and Molnár, 2015). Therefore, this limits the amount of information that such a technique can analyse which in turn affects the identification performance.

The research community has, therefore, proposed a range of statistical methods to overcome the limitation of dynamic port numbers and encrypted traffic approaches in user identification and behaviour profiling based on the application level domain [8, 13]. A reasonable accuracy of up to 85% was achieved by applying statistical features-based methods such as inter-arrival time and packet size to identify users who generated the examined network traffic(Oudah, Ghita and Bakhshi, 2017). The user behaviour profile to be identified from the statistical application levels have noisy traffic. For instance, when a user browses several websites, it is obvious that different patterns would be generated which make the identification of the user more challengeable based on the application level (McDowell, 2013).

In addition, a number of recent studies [9, 16] have used behavioural profiling in identifying computer network users using the DNS and number of connections in addition to statistical method, by collecting the daily DNS queries and connecting the user sessions with reasonable accuracy of up to 75%. However, the accuracy of identified users' traffic based on DNS is affected by the number of days on the observed data as investigated in (Kirchler *et al.*, 2016), which has an accuracy of 73% and 90% with a duration of 65 days and seven days respectively. A variety of studies have examined user behaviour profiling from different perspectives such as identification to distinguishing users [9, 17]. The used techniques are primarily utilised to identify a user is by storing the previous user activities to be able to decide whether or not the examined user is legitimate. However, the use of the behavioural-based technique is by using the observation of how the client communicates with applications from interaction activity such as the size of the packet while uploading a video on YouTube (Alotibi *et al.*, 2015).

As a result, the user behaviour profiling is an appropriate solution in associating with changing of the user behaviour and application over time in a computer network.

To sum up, each method has its limitation and strength based on different circumstances. Obviously, relying on only IP addresses approach to tag individual may not be effective enough in analysing network traffic. This limitation enforces the Internet service provider to examine and analyse larger volumes of raw traffic to identify users, which is suffering from timing consuming and complexity.

### Proposed Method

This study focuses on extracting and analysing and a flow-level features set that allows identifying the user's behaviour activities. The features set extracted was utilised to investigate the users' identification and their daily Internet usage based on the filtered applications sessions. For training

and annotating purposes, used applications were identified based on DNS queries lookup (Internet Systems Consortium, 2005) to be utilised as a feature on the featuring generation process. The raw network traffic was analysed in terms of representing user's daily usage by using a combination of features based on the session, timing and flow DNS filtering. The concept of the session can be described as a group of consecutive flows with shorter inter-arrival time (i.e. the time between two consecutive flows) and the pre-defined threshold between connections.

The threshold value was determined by conducting a preliminary analysis that computes the inter-arrival time distribution among the flows. Accordingly, the session is the measure of the variability of user behaviour changing based on timing resolution bins extracted from start/end time of each application sessions (for instance, 24-hour resolution). This is based on the fact that users' behaviour changes over time in the context of the dynamic changing nature of network applications. In parallel, users' behaviour is also changing as the online interaction's environment change, which develops the problem of these relevant extracted features. The experiment was carried out using a dataset captured from the University of Plymouth, the Centre for Security Communications and Network Research (CSCAN) lab for nine users. The raw network traffic was stored as Netflow records using nfdump (Haag, 2006). The stored flows were pre-processed using Python scripts to filter users based on the MAC/IP address mapping and applications based on DNS queries, as well as create additional statistical features derived from the basic bidirectional NetFlow records. Finally, the dataset was statistically summarised to produce daily user application level records. The newly extracted features were fed into a gradient boosting machine learning algorithm to create a user profile. More details are explained in the next subsections.

### *Data Collection*

The dataset was collected for nine users for a period of 60 days (starting from May 8<sup>th</sup>, 2018 till July 8<sup>th</sup>, 2018) from the student network within the Centre for Security, Communications and Network Research (CSCAN) at Plymouth University, to ensure that the collected data captures most of the user's patterns such as used applications and variability in their behaviour over time. During this period, the participants accessed the Internet through the university network and performed their normal daily routine such as browsing and downloading on the Internet. Participants were not required to do anything but to merely use their device(s) in their typical fashion. The data was collected during their browsing of the internet and was stored in NetFlow file format, together with the MAC/IP mapping to ensure that IP changes due to

DHCP allocation do not affect the accuracy. The top eleven applications were selected based on the statistical procedure, which was computed by implementing the DNS queries keywords for all users to count the connections for each application and choose the top 11 connected and used applications and websites on the lab (i.e., Amazon, Google, Instagram, Facebook, Plymouth, Yahoo, MSN, Unknown, Stack overflow, TeamViewer, and IEEE). Therefore, these applications were added to the session generator for the applications filtering and labelling purposes. Users were filtered by using the MAC address mapping to label the data related to each user. summarises the collected data of the conducted experiment.

Collected Data for Nine Users

User	Flows Count	Duration (days)
1	79,826,114	60
2	20,375,689	60
3	9,625,375	60
4	5,136,927	60
5	15,872,030	60
6	29,818,124	60
7	13,303,973	60
8	11,725,478	60
9	4,906,050	60

### Data pre-processing

The collected data were pre-processed by generate the bidirectional network traffic information. The raw network traffic was generated in several steps regarding getting the most relevant flow-level features to identify the users based on the application sessions and timing resolution criteria. The next subsections explain the undertaken steps to pre-process the raw network traffic to extract desired features.

#### a) Acquiring Raw Network Traffic:

The collected data were initially analysed by nfdump tool to generate the daily raw network traffic for all users in the research centre. In addition, the flow records were expanded to get specific bidirectional NetFlow data records including date start/end time, IP source, IP destination, in pkt (source to destination packets), in byte (source to destination bytes), out pkt (destination to source packets), out byte (destination to source bytes), bps (bits per second), pps (packets per second), and bpp (bytes per packets).

#### b) *Media Accesses Control and IP Source Mapping*

The monitored network for the data collection was maintained by DHCP, therefore the Netflow

collection was accompanied by IP/MAC mapping, to ensure that profiling is allocated to the correct host even if the IP addresses change. The MAC address is unique, and this makes the MAC addresses instead of IPs more reliable to separate the data related to each client for the training purposes. shows a sample of MAC addresses along with its corresponding IP to keep tracking of the IP assignment.

MAC Address and IP Source Mapping

Timestamp	Media Access control	IP source
1526029632	b86b23eb1d7f	192.168.200.170

### c) Domain Name Lookup

The associated domain names are resolved for each Netflow record using a bash script (Internet Software Consortium, 2004); this is in line with the use of DNS queries were in several previous studies for tracking user behaviour and activity [19,17]. The DNS lookup utility (Internet Systems Consortium, 2005) was utilized on bash script to initialise the application name (domain name) for each queried flow, by converting IP destination to domain name. The converted domain name was added as a new attribute (DNS queries) to the Netflow records attributes to be analysed on the next process of this study as shown in . Therefore, the primary aim of using the DNS lookup utility in this study is to determine which flow belongs to the application that facilitates the automated application flow filtering process.

Domain Name Lookup Process

No.	Attribute	Explanation
1	date & time	Date and start /end time
2	IP dst	IP destination
3	in pkt	source to destination packets
4	DNS queries	The DNS query as 'bbc-vip016.cwwtf.bbc.co.uk.'
5	in byte	source to destination bytes
6	out pkt	destination to source packets
7	out byte	destination to source bytes
9	Bps	source to destination bits per second
10	Pps	source to destination packets per second
11	Bpp	source to destination bytes per packets

### 1) Application Flow Filtering Based on the Domain Name

The flows were filtered and separated into groups (applications set) based on pre-defined keywords related to the 11 selected applications (i.e., Amazon, Google, Instagram, Facebook, Plymouth, Yahoo, MSN, Unknown, Stack overflow, TeamViewer, IEEE). Furthermore, DNS query results are classified as unknown if the DNS lookup utility generator could not return any value from the given IP destination address. The applications flow

traffic connections that were filtered and combined in data frames (similar to matrix data object) in terms of representing the usage and automate the way of dealing with the raw network traffic for each client's duration as the data related to each user separated in the previous steps. Furthermore, the filtered data frame is used in the session generator step for the features analysis.

## 2) Session Generator

The filtered applications' data frames are then analysed and divided into sessions using a predefined inter-arrival time threshold, assuming that packets in any flow are relatively uniformly spread over the duration of the flow (Garsva *et al.*, 2014). The flows inter-arrival time is denoted by  $\tau$  (i.e.,  $\tau = \text{the start time of the second flow} - \text{the start time of the first flow}$ ) after converting the date and time to epoch timestamp. The session parameters (Extracted Statistical and Timing Resolution Features0) were calculated based on a flows inter-arrival time threshold based on two conditions: the flows are part of the same session when the  $\tau$  is less than the threshold (i.e., 10 seconds) and the new session starts when the  $\tau$  is higher than the threshold (i.e., 10 seconds). Furthermore, this procedure is applied in all filtered application data frames in order to divide each application to a set of sessions by generating features based on the session concept.

### A. Features Generation Process

The features set generation and their discriminative strength are paramount in maximising the accuracy of the user identification. Two types of features, statistical features and timing-based resolution features were extracted for the dataset.

*1) Extracted Statistical Features:* These features were derived directly from the Netflow records (nfdump) (e.g., source to destination packets and source to destination size of packets as shown in ). Besides, there were other features that were not derived by nfdump, and a calculation was applied to get the complete form of bidirectional flow records data, e.g., destination to source bits per second (bps), destination to source packets per second (pps), destination to source bytes per packets (bpp). Additional features were derived or computed from the above (e.g., Transmitted\_data rate, Received\_to\_transmitted packets). In addition, statistical measures were utilised on the extracted statistical flow session features (i.e., maximum, minimum, mean and median as shown in ).

### Bidirectional Features Sets

NO.	Features	Explanation
-----	----------	-------------

class	User ID	User1, User2, User3	Extracted Statsical Features				24-hours Timing Encoding				
Extracted Statistical Features (Max, Min, Mean, Median)			User	F1	.....	.....	Fn	App	Day	Start_hour	End_hour
1	In pkt	Session Source to destination packet	1	.....	.....	.....	.....	0	0	0	0
2	In byte	Session Source to destination bytes	1	.....	.....	.....	.....	0	0	1	1
3	Out pkt	Session Destination to source packet	1	.....	.....	.....	.....	1	0	.	.
4	Out byte	Session Destination to source bytes	2	.....	.....	.....	.....	2	1	.	.
5	bps	Session source to destination per second	2	.....	.....	.....	.....	4	1	23	23
6	pps	Session source to destination packets per second	Daily user session resolution								
7	bpp	Session source to destination bytes per packet	Daily user session resolution								
8	D2s bps	Session destination to source bytes per second	Daily user session resolution								
9	D2s pps	Session destination to source packets per second	Daily user session resolution								
10	D2s Bpp	Session destination to source bytes per packet	Daily user session resolution								
11	Transmitted data rate	Session transmitted data rate	Daily user session resolution								
12	Received data rate	Session received data rate	Daily user session resolution								
13	Received to transmitted packets	Session Received to transmitted packets	Daily user session resolution								
14	Received to transmitted data	Session received to transmitted data	Daily user session resolution								
Session Timing Resolution Features			Daily User App Level Resolution								
15	Start time	Session start time	Daily user session resolution								
16	End time	Session end time	Daily user session resolution								
17	Number_of_connections	Session Number of connections	Daily user session resolution								
18	Day_of_the_week	Date encoded from (0-7)	Daily user session resolution								
19	Application	Application name encoded (0-10)	Daily user session resolution								
20	Start_hour	Integer encoded from (0-23)	Daily user session resolution								
21	end_hour	Integer encoded from (0-23)	Daily user session resolution								
22	Start/end hour	Start / end hour integer from (0-23) represented on (0-1) timing bins	Daily user session resolution								

ii) *Session Timing Resolution Features*: The timing based features were extracted based on the start and end time of the sessions that are proposed by this study and it includes two types of features relating to the user activity characteristics: session activity and application usage as shown in .

a) *Daily User Session Encoding*: Once the session was generated as defined on and features sets (Extracted Statistical Features, Timing Resolution Features) were extracted for each application's session based on the process explained on the previous sections. Then, the start/end time (hour) was extracted into a separated column as an integer that represents hour (0-23), as shown in . The 24-hour was encoded in terms of combining the start and end timing resolution for the whole sessions related to one application, to represent the daily usage as explained in the next section. Furthermore, the feature that represents applications was encoded into an integer based on the initialised application name to be able to operate with many machines learning which require an input as numeric rather than labels, by converting each categorical feature into numerical (0-10). In addition, the day feature is encoded from (0-7) to represent the day of the week.

Extracted Statsical Features			24-hours Timing Resulation Features									
User	F1	.....	.....	Fn	App	Day	Start hour	End hour	hour (0-9)	hour (10-15)	hour (16-19)	hour (20-23)
1	.....	.....	.....	.....	0	0	0	9	1	0	0	1
1	.....	.....	.....	.....	1	0	10	15	0	1	0	0
1	.....	.....	.....	.....	2	0	16	19	1	0	1	0
1	.....	.....	.....	.....	3	0	20	23	0	0	0	1

Daily User App Level Resolution

#### IV. Evaluation

Gradient boosting is a useful practical supervised machine learning for different predictive tasks, and it can dependably provide more accurate results than the straight single machine learning models. Furthermore, gradient boosting supervised machine learning was utilised to build a series of small decision trees based on the collected data and each tree attempts to correct errors from the previous stage. During the last few years, many practical studies were published, which use decision trees as the base learning for gradient boosting [26, 27]. Furthermore, the algorithm has the ability to optimise any differentiable loss

function by using a gradient descent approach (Yang, 2010). This approach builds the trees sequentially to sum an individual tree consecutively, which provide the best solution under different conditions. In addition, the Z-score was applied to the dataset to normalise the numeric data, excluding the binary bins features for higher accuracy on the end classification model (Kotsiantis, Kanellopoulos and Pintelas, 2006). The data were split randomly into two sets, 70% of the data were used to train the gradient boosting classifier while 30% of the data were used for testing between all user's data. The classifier performance was evaluated with different metrics derived from the four parameters: True positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). The evaluation parameters (accuracy, precision, recall, and F1score) were calculated based on error rates, which are represented in the confusion matrix.

### A. Experimental Results:

The results were shown that concentrating on the flows generic network traffic analysis can produce a significant result in terms of user identification and behaviour profiling. Furthermore, the features were extracted based on the application Flow sessions and timing resolution criteria enhanced the classifier to differentiate between users as a sample belonging to various users in the set. In comparison with previous studies[9, 17], in (Vinupaul *et al.*, 2017) the achieved accuracy was 73% by using flow network analysis approach, while our study achieved up to 80% level of accuracy. Therefore, different aspects affected the accuracy between our study and compared studies. shows the accuracy of all features ranks ranging between 70-80%. The accuracy of rank 2 which represents the second 30 days of the data exceeded the accuracy of rank 1 and rank 3 which represent the first 30 days and all the 60 days of the data. Therefore, rank 1 and rank 3 data were affected by less traffic generated by the experimental lab users (due to a holiday) while rank 2 traffic generated by users were normal and these affect the volume of interactions of the examined users in those ranks. The highest accuracy on rank 2 improves the proposed measurement features that were affected by the periods of collecting data and user access limitation, which was observed on the volume of traffic data between all ranks.

Users' Traffic Classification Results

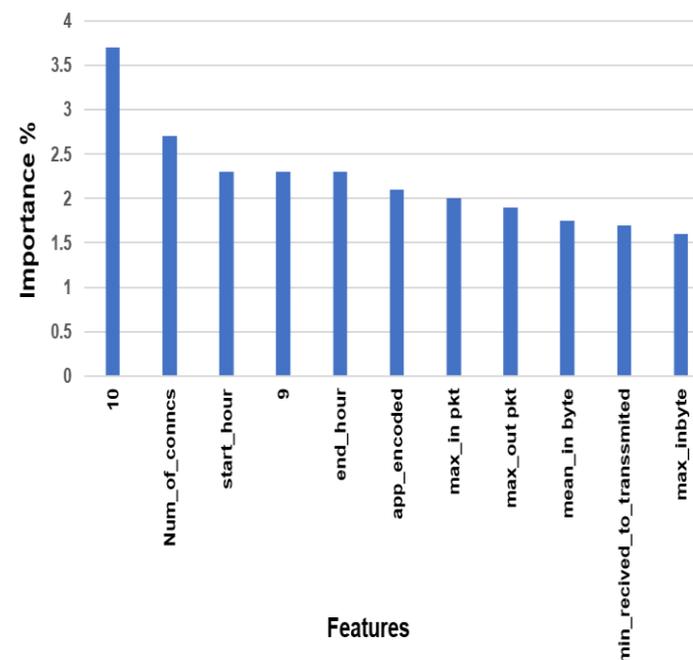
Ranks	No. Users	No. Days	Accuracy	Precision	Recall	F1 score
Rank 1	9	1 <sup>st</sup> /30	72 %	72 %	72 %	71 %
Rank 2	9	2 <sup>nd</sup> /30	80 %	79 %	80 %	79 %
Rank 3	9	60	76 %	77 %	74 %	75 %

The classification comparison was implemented by the gradient boosting using the rank 2 feature set as shown in . The comparison indicates the extracted statistical features and session timing resolution features by employing them to the classifier separately. The session timing resolution features indicated the highest usage score up to 74% compared to the extracted statistical features which were up to 65%. The session timing resolution attributes were scored the highest usage among all users. Also, the rank 2 features were applied to random forest feature importance, which indicated a good performance between all features to identify users.

Classification performance for each feature type

Feature type	Accuracy	Precision	Recall	F1 score
Timing resolution features	76 %	75 %	75 %	74 %
Extracted statistical features	64 %	68 %	63 %	65 %
Both	80 %	79 %	80 %	79 %

Therefore, the top 10 features are represented in , in which the first top 5 features (10, num\_of\_connections, start\_hour, 9, end\_hour, app\_encoded) were scored the highest usage between extracted statistical and proposed session timing resolution features.



Random Forest Feature Importance

### B. Confusion matrix

The most straightforward way to evaluate the performance of the classifier is based on a confusion matrix especially when the model has more than two classes. illustrates a confusion matrix for all users to show the correct and incorrect prediction of each class based on the test data for rank 2 features set. The performance of the classification model is ideally high between all classes ranging from 67-91%. The labels are indicated with users' id from (user1-user9) as illustrated on Table 6 for the predicted and true labels. The highest score of the actual class predict 91% as the TP for User 3, and the FN is 9% which belongs to User 6. Furthermore, User 5 scored the lowest accuracy which is 67% of TP classified samples, there were 33% false negative misclassified recorded for users (1, 3, 6 and 8). Also, User 2 was recorded as the second highest score on the module with 86% TP and 14% misclassified attributed to users (4, 5 and 6).

Confusion Matrix (Features Rank 2)

		Predicted Label									
		1	2	3	4	5	6	7	8	9	
True Label	UI D	1	8	0	0	0	5	4	5	0	5
	2	0	8	0	4	5	5	0	0	0	
	3	0	0	9	0	0	9	0	0	0	
	4	6	0	0	7	6	0	0	1	0	
	5	8	0	8	0	6	8	0	9	0	
	6	4	4	0	9	5	7	0	0	0	
	7	3	3	0	3	0	6	8	4	0	
	8	9	0	0	9	0	0	0	8	0	
	9	1	0	0	5	0	0	0	0	7	

## V. Conclusion and Future Work

The present work proposes a method for user identification and behaviour profiling from generic network traffic. The resulted classification accuracy shows that the proposed features based on application-level flow sessions could be utilised to discriminate among users with an accuracy of up to 80%. A supervised machine-learning algorithm was utilised to evaluate the analysis algorithm with real data collected from the Centre for Security, Communications and Network Research (CSCAN) at Plymouth University to investigate the proposed approach.

Apart from the future work, different timing resolutions and different distribution analysis will be applied on the session's inter-arrival time, to investigate the impact of different thresholds and its effect on system performance. Additionally, more experimental work with different machine-learning algorithms will be utilised to investigate their effect on the proposed method.

## References

- Afridi, M. W., Ali, Toqeer, Alghamdi, T., Ali, Tamleek and Yasar, M. (2018) 'Android application behavioral analysis through intent monitoring', *6th International Symposium on Digital Forensic and Security, ISDFS 2018 - Proceeding*. IEEE, 2018-Janua, pp. 1–8. doi: 10.1109/ISDFS.2018.8355359.
- Alcock, S. and Nelson, R. (2012) 'Libprotoident: Traffic Classification Using Lightweight Packet Inspection Categories and Subject Descriptors', pp. 1–6.
- Algiriyage, N., Jayasena, S. and Dias, G. (2015) 'Web user profiling using hierarchical clustering with improved similarity measure', in *MERCon 2015 - Moratuwa Engineering Research Conference*. IEEE, pp. 295–300. doi: 10.1109/MERCon.2015.7112362.
- Alotibi, G., Li, F., Clarke, N. and Furnell, S. (2015) 'Behavioral-Based Feature Abstraction from Network Traffic', *Iccws 2015-The Proceedings of the 10th International Conference on Cyber Warfare and Security*, (March 2016), pp. 1–9.
- Alotibi, G., Clarke, N., Li, F. and Furnell, S. (2017) 'User profiling from network traffic via novel application-level interactions', in *2016 11th International Conference for Internet Technology and Secured Transactions, ICITST 2016*, pp. 279–285. doi: 10.1109/ICITST.2016.7856712.
- Auld, T., Moore, A. W. and Gull, S. F. (2007) 'Bayesian neural networks for internet traffic classification.', *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 18(1), pp. 223–239. doi: 10.1109/TNN.2006.883010.
- Bakhshi (2017) 'User-Centric Traffic Engineering in Software Defined Networks', in *2016 23rd International Conference on Telecommunications (ICT)*. 2016 23rd International Conference on Telecommunications (ICT), pp. 1–6. Available at: <http://hdl.handle.net/10026.1/8202>.
- Bakhshi, T. and Ghita, B. (2015) 'User Traffic Profiling: In a Software Defined Networking Context', in *2015 Internet Technologies and Applications (ITA)*. Wrexham, pp. 91–97.
- Bakhshi, T. and Ghita, B. (2016a) 'On Internet Traffic Classification: A Two-Phased Machine Learning Approach', *Journal of Computer Networks and Communications*, 2016. doi: 10.1155/2016/2048302.
- Bakhshi, T. and Ghita, B. (2016b) 'Traffic profiling: Evaluating stability in multi-device user environments', *Proceedings - IEEE 30th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2016*, pp. 731–736. doi: 10.1109/WAINA.2016.8.
- Balram, S. and Wilsy, M. (2014) 'User Traffic Profile for Traffic Reduction and Effective Bot C&C

- Detection.’, *IJ Network Security*, 16(1), pp. 46–52. Available at: <http://isrc.asia.edu.tw:8080/contents/ijns-v16-n1/ijns-2014-v16-n1-p46-52.pdf>.
- Banse, C., Herrmann, D. and Federrath, H. (2012) ‘Tracking users on the Internet with behavioral patterns: Evaluation of its practical feasibility’, *IFIP Advances in Information and Communication Technology*, 376 AICT, pp. 235–248. doi: 10.1007/978-3-642-30436-1\_20.
- Bermudez, in and Mellia, M. (2012) ‘Dns to the rescue: Discerning content and services in a tangled web’, *Proceedings of the ...*, pp. 413–426. doi: 10.1145/2398776.2398819.
- Bianco, A., Mardente, G., Mellia, M., Munafò, M. and Muscariello, L. (2009) ‘Web user-session inference by means of clustering techniques’, *IEEE/ACM Transactions on Networking*, 17(2), pp. 405–416. doi: 10.1109/TNET.2008.927009.
- Bivens, a, Palagiri, C., Smith, R. and ... (2002) ‘Network-Based Intrusion Detection Using Neural Networks’, *Intelligent ...*, 12, pp. 1–15. Available at: <http://cyberunited.com/wp-content/uploads/2013/03/INTRUSION-DETECTION-USING-NEURAL-NETWORKS.pdf>.
- Bonald, T. (2015) ‘Traffic Models for User-Level Performance Evaluation in Data Networks’, *2015 27th International Teletraffic Congress*, pp. 107–115. doi: 10.1109/ITC.2015.20.
- Bujlow, T., Carela-Español, V. and Barlet-Ros, P. (2015) ‘Independent comparison of popular DPI tools for traffic classification’, *Computer Networks*. Elsevier, 76, pp. 75–89.
- Bujlow, T., Riaz, T. and Pedersen, J. M. (2012) ‘A method for classification of network traffic based on C5.0 machine learning algorithm’, *2012 International Conference on Computing, Networking and Communications, ICNC’12*, pp. 237–241. doi: 10.1109/ICCNC.2012.6167418.
- Carela-Espanol, V., Barlet-Ros, P. and Solé-Pareta, J. (2009) ‘Traffic classification with sampled netflow’, *Peopleacupcedu*, 33(2), p. 34. Available at: [http://people.ac.upc.edu/pbarlet/reports/netflow\\_classification-techrep.pdf](http://people.ac.upc.edu/pbarlet/reports/netflow_classification-techrep.pdf).
- Chappell, L. (2012) *Wireshark® Network Analysis*. *Wireshark® Network Analysis*.
- Chi, E. H., Rosien, A. and Heer, J. (2002) ‘LumberJack : Intelligent Discovery and Analysis of Web User Traffic Composition’, *Proceedings of WebKDD*, pp. 1–15.
- Cisco (2016) *Cisco IOS NetFlow - Cisco*. Available at: <http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html> (Accessed: 4 July 2017).
- Cisco (2017) *Solutions - Cisco’s 2017 Visual Networking Index (VNI) Infographic - Cisco*. Available at: <http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/vni-infographic.html> (Accessed: 3 July 2017).
- cisco global* (2019). Available at: <https://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html>.
- Cisco VNI Mobile* (2019).
- Cufoglu, A. (2014) ‘User Profiling-A Short Review’, 108(3), pp. 1–9.
- Dainotti, A., Pescapé, A. and Claffy, K. C. (2012) ‘Issues and future directions in traffic classification’, *IEEE Network*, 26(1), pp. 35–40. doi: 10.1109/MNET.2012.6135854.
- Dehghani, F., Movahhedinia, N., Khayyambashi, M. R. and Kianian, S. (2010) ‘Real-Time Traffic Classification Based on Statistical and Payload Content Features’, in *2010 2nd International Workshop on Intelligent Systems and Applications*, pp. 1–4. doi: 10.1109/IWISA.2010.5473467.
- Deri, L., Martinelli, M., Bujlow, T. and Cardigliano, A. (2014) *NDPI: Open-source high-speed deep packet inspection, IWCMC 2014 - 10th International Wireless Communications and Mobile Computing Conference*. doi: 10.1109/IWCMC.2014.6906427.
- Deris Stiawan (2012) ‘Intrusion threat detection from insider attack using learning behavior-based’, *International Journal of the Physical Sciences*, 7(4), pp. 624–637. doi: 10.5897/IJPS11.1381.
- Du, M., Chen, X. and Tan, J. (2013) ‘Online internet traffic identification algorithm based on multistage classifier’, *China Communications*, 10(2), pp. 89–97. doi: 10.1109/CC.2013.6472861.
- Erman, J., Mahanti, A. and Arlitt, M. (2006) ‘Internet Traffic Identification using Machine Learning’, *Global Telecommunications Conference, 2006. GLOBECOM ’06. IEEE*, pp. 1–6. doi: 10.1109/GLOCOM.2006.443.
- Finsterbusch, M., Richter, C., Rocha, E., Müller, J. A. and Hänßgen, K. (2014) ‘A survey of payload-based traffic classification approaches’, *IEEE Communications Surveys and Tutorials*, 16(2), pp. 1135–1156. doi: 10.1109/SURV.2013.100613.00161.
- Garc??a-Dorado, J. L., Finamore, A., Mellia, M., Meo, M. and Munaf??, M. (2012) ‘Characterization of ISP traffic: Trends, user habits, and access technology impact’, *IEEE Transactions on Network and Service Management*, 9(2), pp. 142–155. doi: 10.1109/TNSM.2012.022412.110184.
- Garsva, E., Paulauskas, N., Grazulevicius, G. and Gulbinovic, L. (2014) ‘Packet inter-arrival time distribution in academic computer network’, *Elektronika ir Elektrotechnika*, 20(3), pp. 87–90. doi: 10.5755/j01.eee.20.3.6683.
- Gu, X., Yang, M., Fei, J., Ling, Z. and Luo, J. (2015) ‘A Novel Behavior-Based Tracking Attack for User Identification’, in *2015 Third International Conference on Advanced Cloud and Big Data*, pp. 227–233. doi: 10.1109/CBD.2015.44.
- Haag, P. (2006) *nfdump and NfSen*. Available at: <http://nfdump.sourceforge.net/>.
- Haiyan, Q., Jianfeng, P., Chuan, F. and Rozenblit, J. W. (2007) ‘Behavior analysis-based learning framework for host level intrusion detection’, *Proceedings of the International Symposium and Workshop on Engineering of Computer Based Systems*, pp. 441–447. doi: 10.1109/ECBS.2007.23.
- Heer, J., Heer, J., Chi, E. and Chi, E. (2001)

- ‘Identification of Web User Traffic Composition using Multi-Modal Clustering and Information Scent’, *In Proc. of the Workshop on Web Mining, SIAM Conference on Data Mining*, pp. 51–58.
- Heer, J. and Chi, E. H. (no date) ‘Identification of Web User Traffic Composition using Multi-Modal Clustering and Information Scent’.
- Hong, Y., Huang, C., Nandy, B. and Seddigh, N. (2015) ‘Iterative-Tuning Support Vector Machine For Network Traffic Classification’, pp. 458–466.
- Internet Software Consortium (2004) *host(1) - Linux man page*. Available at: <https://linux.die.net/man/1/host>.
- Internet Systems Consortium, I. (2005) *DNS lookup utility UBUNTo, DNS lookup utility*. Available at: <http://manpages.ubuntu.com/manpages/bionic/man1/host.1.html>.
- Internet world stats* (2019). Available at: <https://www.internetworldstats.com/stats.htm>.
- Iváncsy, R. and Juhász, S. (2007) ‘Analysis of Web User Identification Methods’, *World Academy of Science, Engineering and Technology*, 2(3), pp. 338–345.
- Jiang, H., Moore, A. W., Ge, Z., Jin, S. and Wang, J. (2007) ‘Lightweight Application Classification for Network Management Categories and Subject Descriptors’, in *Proceedings of the 2007 SIGCOMM workshop on Internet network management*. INM’07, pp. 299–304.
- Jin, Y. *et al.* (2012) ‘A Modular Machine Learning System for Flow-Level Traffic Classification in Large Networks’, *ACM Transactions on Knowledge Discovery from Data*, 6(1), pp. 1–34. doi: 10.1145/2133360.2133364.
- Kabir, S., Mudur, S. P. and Shiri, N. (2012) ‘Capturing browsing interests of users into web usage profiles’, *AAAI Workshop - Technical Report*, WS-12-09, pp. 18–25.
- Kamesh and Sakthi Priya, N. (2014) ‘Security enhancement of authenticated RFID generation’, *International Journal of Applied Engineering Research*, 9(22), pp. 5968–5974. doi: 10.1002/sec.
- Karagiannis, T., Papagiannaki, K. and Faloutsos, M. (2005) ‘BLINC: multilevel traffic classification in the dark’, *ACM SIGCOMM Computer Communication Review*, 35(4), pp. 229–240. doi: <http://doi.acm.org/10.1145/1080091.1080119>.
- Kihl, M. and Odling, P. (2010) ‘Traffic analysis and characterization of Internet user behavior’, ... *and Control Systems ...*, pp. 224–231. Available at: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5676633](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5676633).
- Kirchler, M., Herrmann, D., Lindemann, J. and Kloft, M. (2016) ‘Tracked Without a Trace: Linking Sessions of Users by Unsupervised Learning of Patterns in Their DNS Traffic’, in *the 2016 ACM Workshop on Artificial Intelligence and Security*, pp. 23–34. doi: 10.1145/2996758.2996770.
- Kotsiantis, S. B., Kanellopoulos, D. and Pintelas, P. E. (2006) ‘Data preprocessing for supervised learning’, *International Journal of Computer Science*, 1(2), pp. 111–117. doi: 10.1080/02331931003692557.
- Kounavis, M. E., Kumar, A., Vin, H., Yavatkar, R. and Campbell, A. T. (2004) ‘Directions in Packet Classification for Network Processors’, *Volume 2 in The Morgan Kaufmann Series in Computer Architecture and Design*, pp. 273–298.
- Kumar, G., Kumar, K. and Sachdeva, M. (2010) ‘The use of artificial intelligence based techniques for intrusion detection: A review’, *Artificial Intelligence Review*, 34(4), pp. 369–387. doi: 10.1007/s10462-010-9179-5.
- Kumar, R. (2014) ‘Machine Learning based Traffic Classification using Low Level Features and Statistical Analysis’, 108(12), pp. 6–13.
- Li, B., Springer, J., Bebis, G. and Hadi Gunes, M. (2013) ‘A survey of network flow applications’, *Journal of Network and Computer Applications*, pp. 567–581. doi: 10.1016/j.jnca.2012.12.020.
- Lim, S. Y. and Jones, A. (2008) ‘Network Anomaly Detection System: The State of Art of Network Behaviour Analysis’, *2008 International Conference on Convergence and Hybrid Information Technology*, pp. 459–465. doi: 10.1109/ICHIT.2008.249.
- Liu, C. and Wu, J. (2013) ‘Fast deep packet inspection with a dual finite automata’, *IEEE Transactions on Computers*, 62(2), pp. 310–321. doi: 10.1109/TC.2011.231.
- Lucas, M. W. (Michael W. (2010) *Network flow analysis*. No Starch Press. Available at: [https://books.google.co.uk/books?id=5MDucc0LwiUC&pg=PA35&lpg=PA35&dq=softflowd+manual&source=bl&ots=BDQp4yoNfc&sig=BhmOPi5xtFxiKKnfh0R4RItN6rl&hl=en&sa=X&ved=0ahUKewia\\_sfTg5PUA hWIKsAKHa00BBEQ6AEINDAC#v=onepage&q=softflowd manual&f=false](https://books.google.co.uk/books?id=5MDucc0LwiUC&pg=PA35&lpg=PA35&dq=softflowd+manual&source=bl&ots=BDQp4yoNfc&sig=BhmOPi5xtFxiKKnfh0R4RItN6rl&hl=en&sa=X&ved=0ahUKewia_sfTg5PUA hWIKsAKHa00BBEQ6AEINDAC#v=onepage&q=softflowd manual&f=false) (Accessed: 28 May 2017).
- Malott, L. and Chellappan, S. (2014) ‘Investigating the fractal nature of individual user netflow data’, *Proceedings - International Conference on Computer Communications and Networks, ICCCN*. doi: 10.1109/ICCCN.2014.6911837.
- McDowell, C. M. (2013) *Creating Profiles From User Network Behavior*. Available at: [http://calhoun.nps.edu/bitstream/handle/10945/37673/13Sep\\_McDowell\\_Chad.pdf?sequence=1](http://calhoun.nps.edu/bitstream/handle/10945/37673/13Sep_McDowell_Chad.pdf?sequence=1).
- Medhi, D. (no date) *tcpdump and libpcap latest release*. The Tcpdump Group. Available at: <http://www.tcpdump.org/#latest-release> (Accessed: 10 June 2017).
- Megyesi, P. and Molnr, S. (2012) ‘Finding typical internet user behaviors’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7479 LNCS, pp. 321–327. doi: 10.1007/978-3-642-32808-4\_29.
- Melnikov, N. (2010) ‘Cybermetrics: User Identification through Network Flow Analysis’, *Ifip International Federation For Information Processing*, pp. 167–170.
- Melnikov, N. and Schönwälder, J. (2010) ‘User identification based on the analysis of network flow patterns’, *Cnds.Eecs.Jacobs-University.De*. Available at: <http://cnds.eecs.jacobs-university.de/courses/nds-2009/melnikov-report.pdf>.
- Mukkamala, S., Janoski, G. and Sung, a. (2002)

- 'Intrusion detection using neural networks and support vector machines', *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN '02*, pp. 1702–1707. doi: 10.1109/IJCNN.2002.1007774.
- Nascimento, Z., Sadok, D., Fernandes, S. and Kelner, J. (2014) 'Multi-objective optimization of a hybrid model for network traffic classification by combining machine learning techniques', *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 2116–2122. doi: 10.1109/IJCNN.2014.6889935.
- Nguyen, T. T. T. and Armitage, G. (2008) 'A survey of techniques for internet traffic classification using machine learning', *Communications Surveys & Tutorials, IEEE*, 10(4), pp. 56–76. doi: 10.1109/SURV.2008.080406.
- No, I. and Jamuna, A. (2013) 'Available Online at www.ijarcs.info Survey of Traffic Classification using Machine Learning', 4(4), pp. 65–72.
- Oh, S. H. and Lee, W. S. (2003) 'An anomaly intrusion detection method by clustering normal user behavior', *Computers and Security*, 22(7), pp. 596–612. doi: 10.1016/S0167-4048(03)00710-7.
- Oliveira, E. de (2011) *Methodologies for Traffic Profiling in Communication Networks*.
- Oudah, H., Ghita, B., Bakhshi, T., Alruban, A. and Walker, D. J. (2019) 'Using Burstiness for Network Applications Classification', *Journal of Computer Networks and Communications*, 2019(August), pp. 1–10. doi: 10.1155/2019/5758437.
- Oudah, H., Ghita, B. and Bakhshi, T. (2017) 'Network Application Detection Using Traffic Burstiness', in *WorldCIS*.
- Paredes-Oliva, I., Castell-Uroz, I., Barlet-Ros, P., Dimitropoulos, X. and Solé-Pareta, J. (2012) 'Practical anomaly detection based on classifying frequent traffic patterns', *Proceedings - IEEE INFOCOM*, pp. 49–54. doi: 10.1109/INFCOMW.2012.6193518.
- Park, N. H., Oh, S. H. and Lee, W. S. (2010) 'Anomaly intrusion detection by clustering transactional audit streams in a host computer', *Information Sciences*. Elsevier Inc., 180(12), pp. 2375–2389. doi: 10.1016/j.ins.2010.03.001.
- PéterMegyesi, Szabó, G. and Molnár, S. (2015) 'User behavior based traffic emulator: A framework for generating test data for DPI tools', *Computer Networks*, 92, pp. 41–54. doi: 10.1016/j.comnet.2015.09.026.
- Phan, M. C., Sun, A. and Tay, Y. (2017) 'Cross-Device User Linking: URL, Session, Visiting Time, and Device-log Embedding', *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 933–936. doi: 10.1145/3077136.3080682.
- Piskac, P. and Novotny, J. (2011) 'Network Traffic Classification Based on Time Characteristics Analysis', *Masaryk University Faculty of Informatics*, (January). Available at: [http://is.muni.cz/th/173297/fi\\_r/thesis.pdf](http://is.muni.cz/th/173297/fi_r/thesis.pdf).
- Plonka, D. and Barford, P. (2011) 'Flexible Traffic and Host Profiling via DNS Rendezvous', *Workshop SATIN*.
- Potdar, K., S., T. and D., C. (2017) 'A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers', *International Journal of Computer Applications*, 175(4), pp. 7–9. doi: 10.5120/ijca2017915495.
- Qadeer, M. A., Zahid, M., Iqbal, A. and Siddiqui, M. R. (2010) 'Network Traffic Analysis and Intrusion Detection Using Packet Sniffer', *Communication Software and Networks, 2010. ICCSN '10. Second International Conference on*, pp. 313–317. doi: 10.1109/ICCSN.2010.104.
- Qin, T., Guan, X., Wang, C. and Liu, Z. (2014) 'MUCM: Multilevel User Cluster Mining Based on Behavior Profiles for Network Monitoring', *IEEE Systems Journal*, pp. 1–12. doi: 10.1109/JSYST.2014.2350019.
- Rossi, D. and Valenti, S. (2010) 'Fine-grained traffic classification with netflow data', *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference on ZZZ - IWCMC '10*, p. 479. doi: 10.1145/1815396.1815507.
- Ryan, J., Lin, M. J. and Miikkulainen, R. (1998) 'Intrusion Detection with Neural Networks', *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10*, pp. 943–949.
- Shaikh, Z. A., Science, C. and Science, C. (no date) 'An Overview of Network Traffic Classification Methods'.
- Singh, H. (2015) 'Performance Analysis of Unsupervised Machine Learning Techniques for Network Traffic Classification', *2015 Fifth International Conference on Advanced Computing & Communication Technologies*, pp. 401–404. doi: 10.1109/ACCT.2015.54.
- Sinha, A., Mitchell, K. and Medhi, D. (2003) 'Flow-level upstream traffic behavior in broadband access networks: DSL versus broadband fixed wireless', *Proceedings of the 3rd IEEE Workshop on IP Operations and Management, IPOM 2003*, pp. 135–141. doi: 10.1109/IPOM.2003.1251235.
- Suznjevic, M., Skorin-kapov, L. and Humar, I. (2014) 'User Behavior Detection Based on Statistical Traffic Analysis for Thin client Services', 2, pp. 247–256. doi: 10.1007/978-3-319-05948-8.
- Szabó, G., Turányi, Z., Toka, L., Molnár, S. and Santos, A. (2011) 'Automatic Protocol Signature Generation Framework for Deep Packet Inspection', *Proc. of ICST*, pp. 291–299. doi: 10.4108/icst.valuetools.2011.245606.
- Tao, M., Chun, Y. and Juan, C. (no date) 'Profiling and Identifying Users' Activities With Network Traffic Analysis', pp. 503–506.
- Therriault, K., Vukelich, D., Farrell, W., Kong, D. and Lowry, J. (2015) 'Network Traffic Analysis Using Behavior-Based Clustering', (November).
- Ulliac, A. and Ghita, B. V. (2010) 'Non-intrusive identification of peer-to-peer traffic', in *3rd Int. Conf. on Communication Theory, Reliability, and Quality of Service, CTRQ 2010, Includes MOPAS 2010: 1st Int. Conf. on Models and Ontology-Based Design of Protocols, Architecture and Services*, pp. 116–121. doi: 10.1109/CTRQ.2010.27.
- Veres, S. and Ionescu, D. (2009) 'Measurement-Based Traffic Characterization for Web 2.0 Applications', in *Scenario*, pp. 5–7.

- Vinupaul, M. V., Bhattacharjee, R., Rajesh, R. and Kumar, G. S. (2017) 'User characterization through network flow analysis', in *Proceedings of the 2016 International Conference on Data Science and Engineering, ICDSE 2016*, pp. 1–6. doi: 10.1109/ICDSE.2016.7823965.
- Wagner, C., State, R., Engel, T. and Learning, M. (2011) 'Machine Learning Approach for IP-Flow Record Anomaly Detection To cite this version : Machine Learning Approach for IP-Flow Record Anomaly Detection', *International Conference on Research in Networking*, pp. 28–39.
- Wang, J. H., An, C. and Yang, J. (2011) 'A study of traffic, user behavior and pricing policies in a large campus network', *Computer Communications*, 34, pp. 1922–1931. doi: 10.1016/j.comcom.2011.05.009.
- Wang, S., State, R., Ourdane, M. and Engel, T. (2011) 'Mining NetFlow Records for Critical Network Activities'.
- Williams, N., Zander, S. and Armitage, G. (2006) 'A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification', *ACM SIGCOMM Computer Communication Review*, 36(5), p. 5. doi: 10.1145/1163593.1163596.
- Xu, K., Zhang, Z. and Bhattacharyya, S. (2008) 'Internet Traffic Behavior Profiling for Network', *IEEE/ACM Transactions on Networking*, 16(6), pp. 1241–1252. doi: 10.1109/TNET.2007.911438.
- Xue, Y. and Dong, Y. (2013) 'Harvesting unique characteristics in packet sequences for effective application classification', *2013 IEEE Conference on Communications and Network Security (CNS)*, pp. 341–349. doi: 10.1109/CNS.2013.6682724.
- Yang, B., Hou, G., Ruan, L., Xue, Y. and Li, J. (2011) 'SMILER: Towards Practical Online Traffic Classification', *2011 ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems*, pp. 178–188. doi: 10.1109/ANCS.2011.34.
- Yang, J. *et al.* (2015) 'Characterizing user behavior in mobile internet', *IEEE Transactions on Emerging Topics in Computing*, 3(1), pp. 95–106. doi: 10.1109/TETC.2014.2381512.
- Yang, Y. (2010) 'Web user behavioral profiling for user identification', *Decision Support Systems*, 49(3), pp. 261–271. doi: 10.1016/j.dss.2010.03.001.
- Zhang, C. X., Zhang, J. S. and Zhang, G. Y. (2008) 'An efficient modified boosting method for solving classification problems', *Journal of Computational and Applied Mathematics*, 214(2), pp. 381–392. doi: 10.1016/j.cam.2007.03.003.
- Zhang, F., He, W., Liu, X. and Bridges, P. G. (2011) 'Inferring users' online activities through traffic analysis', *Proceedings of the fourth ACM conference on Wireless network security - WiSec '11*, p. 11. doi: 10.1145/1998412.1998425.
- Zhang, H., Lu, G., Qassrawi, M. T., Zhang, Y. and Yu, X. (2012) 'Feature selection for optimizing traffic classification', *Computer Communications*. Elsevier B.V., 35(12), pp. 1457–1471. doi: 10.1016/j.comcom.2012.04.012.
- Zhang, J., Xiang, Y., Zhou, W. and Wang, Y. (2013) 'Unsupervised traffic classification using flow statistical properties and IP packet payload', *Journal of Computer and System Sciences*. Elsevier Inc., 79(5), pp. 573–585. doi: 10.1016/j.jcss.2012.11.004.

# Ethical approval documentation

## PLYMOUTH UNIVERSITY FACULTY OF SCIENCE AND ENGINEERING

### Research Ethics Committee

#### APPLICATION FOR ETHICAL APPROVAL OF RESEARCH INVOLVING HUMAN PARTICIPANTS

**All applicants should read the guidelines which are available via the following link:**

<https://staff.plymouth.ac.uk/scienv/humanethics/intranet.htm>

This is a WORD document. Please complete in WORD and extend space where necessary.

*All applications must be word processed. Handwritten applications **will** be returned.*

***Please submit with interview schedules and/or questionnaires appropriately.***

*Postgraduate and Staff must submit a signed copy to*  
[SciEngHumanEthics@plymouth.ac.uk](mailto:SciEngHumanEthics@plymouth.ac.uk)

*Undergraduate students should contact their School Representative of the Science and Engineering Research Ethics Committee or dissertation advisor prior to completing this form to confirm the process within their School.*

***School of Computing, Electronics and Mathematics undergraduate students – please submit to [SciEngHumanEthics@plymouth.ac.uk](mailto:SciEngHumanEthics@plymouth.ac.uk) with your project supervisor copied in.***

---

### **1. TYPE OF PROJECT**

#### **1.1 What is the type of project? (Put an X next to one only)**

**STAFF should put an X next to one of the three options below:**

Specific project

Thematic programme of research

Practical / Laboratory Class

#### **1.2 Put an X next to one only**

**POSTGRADUATE STUDENTS should put an X next to one of the options below:**

Taught Masters Project

M.Phil / PhD by research

**UNDERGRADUATE STUDENTS** should put an X next to one of the options below:

Student research project

Practical / Laboratory class where you are acting as the experimenter

## 2. APPLICATION

<b>2.1 TITLE of Research project</b>	
User Profiling Based on Network Application Traffic Monitoring	
<b>2.2 General summary of the proposed research for which ethical clearance is sought, briefly outlining the aims and objectives and providing details of interventions/procedures involving participants (no jargon)</b>	
<p><u>Aim</u> To identify and classify users based on applications behaviour activities usage of the Internet and identify the user behaviour profile</p> <p><u>Methodology</u> A monitoring machine will be setup to capture the Internet traffic for a set of users. The capture will include only the Ethernet, IP, and TCP packet headers, but no application data. The traffic will be analysed using a set of automated software scripts to determine whether individual hosts can be identified based on the user activity, but with no attempt to either physically identify the users beyond an index/identifier or the IP address.</p>	
<b>2.3 Physical site(s) where research will be carried out</b>	
On site or in the office of the institutions involved	
<b>2.4 External Institutions involved in the research (e.g. other university, hospital, prison etc.)</b>	
No	
<b>2.5 Name, telephone number, e-mail address and position of lead person for this project (plus full details of Project Supervisor if applicable)</b>	
1- Faisal Shaman	<a href="mailto:faisal.shaman@plymouth.ac.uk">faisal.shaman@plymouth.ac.uk</a> 07478753933
2- Bogdan Ghita	<a href="mailto:bogdan.ghita@plymouth.ac.uk">bogdan.ghita@plymouth.ac.uk</a> 00441752586237
3- David Lancaster	<a href="mailto:david.lancaster@plymouth.ac.uk">david.lancaster@plymouth.ac.uk</a>
4- Nathan Clark	<a href="mailto:n.clarke@plymouth.ac.uk">n.clarke@plymouth.ac.uk</a>
<b>2.6 Start and end date for research for which ethical clearance is sought (NB maximum period is 3 years)</b>	
Start date: 01/04/2018	End date: 30/09/2018
<b>2.7 Has this same project received ethical approval from another Ethics Committee?</b>	
Delete as applicable:	No
<b>2.8 If yes, do you want Chairman's action?</b>	
Delete as applicable:	No
<b>If yes, please include other application and approval letter and STOP HERE. If no, please continue</b>	

## 3. PROCEDURE

<b>3.1 Describe procedures that participants will engage in, Please do not use jargon</b>
---

The participants will access the Internet as they normally do; network traffic will be captured in the background (only network information, not content/application data). The resulting dataset will subsequently be analysed using a set of automated scripts.
<b>3.2 How long will the procedures take? Give details</b>
The procedure will be transparent to the user, as traffic capture will take place in the background.
<b>3.3 Does your research involve deception?</b>
Delete as applicable: <b>No</b>
<b>3.4 If yes, please explain why the following conditions apply to your research:</b>
<b>a) Deception is completely unavoidable if the purpose of the research is to be met</b>
<b>b) The research objective has strong scientific merit</b>
<b>c) Any potential harm arising from the proposed deception can be effectively neutralised or reversed by the proposed debriefing procedures (see section below)</b>
<b>3.5 Describe how you will debrief your participants</b>
The participants will browse normally some applications through the Internet and the nfcapd /softflowd tool will capture the traffic without any personal data.
<b>3.6 Are there any ethical issues (e.g. sensitive material)?</b>
Delete as applicable: <b>No</b>
<b>3.7 If yes, please explain. You may be asked to provide ethically sensitive material. See also section 11</b>

#### 4. NON-VULNERABLE ADULTS

<b>4.1 Are some or all of the participants non-vulnerable adults?</b>
Yes
<b>4.2 Inclusion / exclusion criteria</b>
Participants who are 18 years old and above, agree and understand all procedure are able to take part in this study.
<b>4.3 How will participants give informed consent?</b>
Participants will be asked to complete a consent form giving permission for their activity to be used in this study.
<b>4.4 Consent form(s) attached</b>
Yes
<b>If no, why not?</b>
<b>4.5 Information sheet(s) attached</b>
Yes
<b>If no, why not?</b>
<b>4.6 How will participants be made aware of their right to withdraw at any time?</b>
It will be stated in the briefing along with the consent form that participants have the right to withdraw at any stage up to the completion of the data collection process.
<b>4.7 How will confidentiality be maintained, including archiving / destruction of primary data where appropriate, and how will the security of the data be maintained?</b>
Participants will be informed that their metadata will be anonymous, securely stored within the Centre for Security, Communications and Network Research (CSCAN). They will also be only used for the purpose stated in the briefing. In accordance with Plymouth University guidelines, the data will be stored for ten years. Once the ten-year time period is reached, the data will be securely destroyed.

#### 5. MINORS <16 YEARS

<b>5.1 Are some or all of the participants under the age of 16?</b>
No
<b>If yes, please consult special guidelines for working with minors. If no, please continue.</b>
<b>5.2 Age range(s) of minors</b>
<b>5.3 Inclusion / exclusion criteria</b>
<b>5.4 How will minors give informed consent? Please tick appropriate box and explain (See guidelines)</b>
N/A
<b>5.5 Consent form(s) for minor attached</b>
N/A
<b>If no, why not?</b>
N/A

<b>6.6 Information sheet(s) for minor attached</b>
N/A
<i>If no, why not?</i>
N/A
<b>6.7 Consent form(s) for parent / legal guardian attached</b>
N/A
<i>If no, why not?</i>
N/A
<b>6.8 Information sheet(s) for parent / legal guardian attached</b>
N/A
<i>If no, why not?</i>
N/A
<b>6.9 How will minors be made aware of their right to withdraw at any time?</b>
N/A
<b>6.10 How will confidentiality be maintained, including archiving / destruction of primary data where appropriate, and how will the security of the data be maintained?</b>
N/A

## **6. MINORS 16-18 YEARS OLD**

<b>6.1 Are some or all of the participants between the ages of 16 and 18?</b>
No
<i>If yes, please consult special guidelines for working with minors. If no, please continue.</i>
<b>6.2 Inclusion / exclusion criteria</b>
N/A
<b>6.3 How will minors give informed consent? (See guidelines)</b>
N/A
<b>6.4 Consent form(s) for minor attached</b>
N/A
<i>If no, why not?</i>
N/A
<b>6.5 Information sheet(s) for minor attached</b>
N/A
<i>If no, why not?</i>
N/A
<b>6.6 Consent form(s) for parent / legal guardian attached</b>
N/A
<i>If no, why not?</i>
N/A
<b>6.7 Information sheet(s) for parent / legal guardian attached</b>
N/A
<i>If no, why not?</i>
N/A
<b>6.8 How will minors be made aware of their right to withdraw at any time?</b>
N/A
<b>6.9 How will confidentiality be maintained, including archiving / destruction of primary data where appropriate, and how will the security of the data be maintained?</b>
N/A

## **7. VULNERABLE GROUPS**

<b>7.1 Are some or all of the participants vulnerable? (See guidelines)</b>
---

<b>No</b>
<i>If yes, please consult special guidelines for working with vulnerable groups. If no, please continue.</i>
<b>7.2 Describe vulnerability (apart from possibly being a minor)</b>
N/A
<b>7.3 Inclusion / exclusion criteria</b>
N/A
<b>7.4 How will participants give informed consent?</b>
N/A
<b>7.5 Consent form(s) for vulnerable person attached</b>
N/A
<i>If no, why not?</i>
N/A
<b>7.6 Information sheet(s) for vulnerable person attached</b>
N/A
<i>If no, why not?</i>
N/A
<b>7.7 Consent form(s) for parent / legal guardian attached</b>
N/A
<i>If no, why not?</i>
N/A
<b>7.8 Information sheet(s) for parent / legal guardian attached</b>
N/A
<i>If no, why not?</i>
N/A
<b>7.9 How will participants be made aware of their right to withdraw at any time?</b>
N/A
<b>7.10 How will confidentiality be maintained, including archiving / destruction of primary data where appropriate, and how will the security of the data be maintained?</b>
N/A

## 8. EXTERNAL CLEARANCES

**Investigators working with children and vulnerable adults legally require clearance from the Disclosure and Barring Service (DBS)**

<b>8.1 Do ALL experimenters in contact with children and vulnerable adults have <u>current</u> DBS clearance? Please include photocopies.</b>
<b>8.2 If your research involves external institutions (school, social service, prison, hospital etc) please provide cover letter(s) from institutional heads permitting you to carry out research on their clients, and where applicable, on their site(s). Are these included?</b>
N/A
<i>If not, why not?</i>
N/A

--

**9. PHYSICAL RISK ASSESSMENT**

<b>9.1 Will participants be at risk of physical harm (e.g. from electrodes, other equipment)? (See guidelines)</b>
No
<b>9.2 If yes, please describe</b>
N/A
<b>9.3 What measures have been taken to minimise risk? Include risk assessment proformas which has been signed by the Head of Department</b>
N/A
<b>9.4 How will you handle participants who appear to have been harmed?</b>
N/A

**10. PSYCHOLOGICAL RISK ASSESSMENT**

<b>10.1 Will participants be at risk of psychological harm (e.g. viewing explicit or emotionally sensitive material, being stressed, recounting traumatic events)? (See guidelines)</b>
No
<b>10.2 If yes, please describe</b>
N/A
<b>10.3 What measures have been taken to minimise risk?</b>
N/A
<b>10.4 How will you handle participants who appear to have been harmed?</b>
N/A

**11. RESEARCH OVER THE INTERNET**

<b>11.1 Will research be carried out over the internet?</b>
No
<b>11.2 If yes, please explain protocol in detail, explaining how informed consent will be given, right to withdraw maintained, and confidentiality maintained. Give details of how you will guard against abuse by participants or others (see guidelines)</b>

**12. CONFLICTS OF INTEREST & THIRD-PARTY INTERESTS**

<b>12.1 Do any of the experimenters have a conflict of interest? (See guidelines)</b>
No
<b>12.2 If yes, please describe</b>
N/A
<b>12.3 Are there any third parties involved? (See guidelines)</b>
N/A
<b>12.4 If yes, please describe</b>
N/A
<b>12.5 Do any of the third parties have a conflict of interest?</b>
N/A

<b>12.6 If yes, please describe</b>
N/A

**13. ADDITIONAL INFORMATION**

<b>13.1 [Optional] Give details of any professional bodies whose ethical policies apply to this research</b>
N/A
<b>13.2 [Optional] Please give any additional information that you wish to be considered in this application</b>
N/A

**14. ETHICAL PROTOCOL & DECLARATION**

To the best of our knowledge and belief, this research conforms to the ethical principles laid down by the University of Plymouth and by any professional body specified in section 14 above.

This research conforms to the University’s Ethical Principles for Research Involving Human Participants with regard to openness and honesty, protection from harm, right to withdraw, debriefing, confidentiality, and informed consent

**Sign below where appropriate:**

**STAFF / RESEARCH POSTGRADUATES**

	<b>Print Name</b>	<b>Signature</b>	<b>Date</b>
Principal Investigator:	Faisal shaman		
Other researchers:	Dr. Bogdan Ghita		
	Dr. David Lancaster		
	Prof. Nathan Clarke		

**Staff and Research Postgraduates should email the completed and signed copy of this form to Paula Simson.  
UG Students**

<b>Date</b>	<b>Print Name</b>	<b>Signature</b>	
Student: -----	Faisal Shaman	-----	----
Supervisor / Advisor: -----	Dr. Bogdan Ghita	-----	----
-----	Dr. David Lancaster	-----	----
-----	Prof. Nathan Clarke	-----	----

**Undergraduate students should pass on the completed and signed copy of this form to their School Representative on the Science and Engineering Human Ethics Committee.**

**Signature** **Date**

**Faculty of Science and Engineering Research Ethics Committee List of School  
Representatives**

School of Geography, Earth and Environmental Sciences	Dr Sanzidur Rahman Dr Kim Ward
School of Biological and Marine Sciences	Dr Gillian Glegg (Chair)  Dr Victor Kuri
School of Biomedical and Healthcare Sciences	Dr David J Price
School of Engineering	Dr Liz Hodgkinson
School of Computing, Electronics & Mathematics	Dr Mark Dixon Dr Yinghui Wei
External Representative	Prof Linda La Velle
Lay Member	Rev. David Evans

Committee Secretary: Mrs Paula Simson

email: paula.simson@plymouth.ac.uk

tel: 01752 584503

**PLYMOUTH UNIVERSITY**

**FACULTY OF SCIENCE AND ENGINEERING**

**Human Ethics Committee Sample Consent Form**

**CONSENT TO PARTICIPATE IN RESEARCH PROJECT / PRACTICAL STUDY**

---

Name of Principal Investigator

Faisal Shaman

---

Title of Research

User Profiling Based on Network Application Traffic Monitoring

---

Brief statement of purpose of work

The main purpose of this work is to identify and classify users based on Internet traffic application usage based on statistical approach and profiling the behaviour activities. So, the data will have captured while the participants work on the Internet applications and for one hour for each application.

The participants have the right to withdraw at any stage upon until the completion of the data collection process.

---

The objectives of this research have been explained to me.

I understand that I am free to withdraw from the research at any stage and ask for my data to be destroyed if I wish.

I understand that my anonymity is guaranteed, unless I expressly state otherwise.

I understand that the Principal Investigator of this work will have attempted, as far as possible, to avoid any risks, and that safety and health risks will have been separately assessed by appropriate authorities (e.g. under COSHH regulations)

Under these circumstances, I agree to participate in the research.

Name: .....

Signature: .....  
.....

Date:

PLYMOUTH UNIVERSITY

FACULTY OF SCIENCE AND ENGINEERING

RESEARCH INFORMATION SHEET

---

Name of Principal Investigator

Faisal Shaman

---

Title of Research

User Profiling Based on Network Application Traffic Monitoring

---

Aim of research

The aim of this project is to propose and investigate novel mechanisms to define application and user behaviour as seen through the generated network traffic. The project is divided into the following distinct stages:

- 1- The project will review prior research, identifying means of recording network, characterizing traffic, and consider NetFlow as a possible data collection method.
- 2- New traffic metrics will be defined for user traffic profiling and recording. To accurately describe the user behavior, the project will consider parameters such as, defining characteristics for a number of typical users considering timing and patterns for user events as part of a network application session.
- 3- Using an extended set of traffic features, machine learning techniques will be used to derive a user behavior traffic profiling system that will be validated and evaluated against a number of applications have been used by the user.
- 4- The project will further extend and explore the proposed traffic profiling design to characterize typical user behaviour that may aid administrators in making formal decision by using the user behavior traffic profile.

## Description of procedure

The participants will access the Internet through the university network and browsing limited applications that the researcher is identified. The data will be collected during their browsing and save it in nfcapd (netflow) file format.

## Description of risks

At no stage will any personally identifiable information be seen by any individual neither the researchers nor on any publication. The captured data will be stored after being converted to measurement features. All of the information will be treated confidentially, and data will be anonymous during the collection, storage and publication of research material.

After the participants finish their work, they do delete by their self the history of the applications that they were visiting them, so the researcher cannot be able to access their data

## Benefits of proposed research

User profiling based on network traffic monitoring can be considered as an initial task of analysing different patterns of generic network traffic information to address the changing on the user behaviour based on application usage to manage different tasks such as policing, traffic management, and enforcing the policy of the organization.

## Right to withdraw

You have the right to withdraw at any time without giving a reason. Your data will be removed and securely deleted.

If you are dissatisfied with the way the research is conducted, please contact the principal investigator in the first instance: Faisal Shaman, A328, Portland Square Building, Plymouth University. Email: [faisal.shaman@plymouth.ac.uk](mailto:faisal.shaman@plymouth.ac.uk) Telephone number [01752 586251], Mobile number [07435395339]. If you feel the problem has not been resolved, please contact the secretary to the Faculty of Science and Engineering Human Ethics Committee: Mrs Paula Simson 01752 584503.

**SAMPLE CONSENT FORM FOR PARENT/LEGAL GUARDIAN**

**PLYMOUTH UNIVERSITY**

**FACULTY OF SCIENCE AND ENGINEERING**

**Human Ethics Committee Sample Consent Form**

**CONSENT TO PARTICIPATE IN RESEARCH PROJECT / PRACTICAL  
STUDY**

---

**Name of Principal Investigator**

Faisal Shaman

---

**Title of Research**

User Profiling Based on Network Application Traffic Monitoring

---

**Brief statement of purpose of work**

The main purpose of this work is to analyse the generic network traffic by classify user behaviour activity based on statistical approach and create user behaviour profile. So, the data will capture while the participants work on the Internet applications and for half an hour for each application.

---

**I am the \*parent /legal guardian of**

---

The objectives of this research have been explained to me.

I understand that \*she/he is free to withdraw from the research at any stage and ask for \*his/her data to be destroyed if I wish.

I understand that \*his/her anonymity is guaranteed, unless I expressly state otherwise.

I understand that the Principal Investigator of this work will have attempted, as far as possible, to avoid any risks, and that safety and health risks will have been separately assessed by appropriate authorities (e.g. under COSHH regulations)

Under these circumstances, I agree for him/her to participate in the research.

*\* delete as*

*appropriate*

Name: .....

Signature: .....

.....

Date:

**RESEARCH  
WITH  
PLYMOUTH  
UNIVERSITY**

14 February 2018

**CONFIDENTIAL**

Faisal Shaman  
School of Computing, Electronics and Mathematics

Dear Faisal

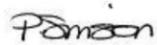
***Ethical Approval Application***

Thank you for submitting the ethical approval form and details concerning your project:

***User Profiling Based on Network Application Traffic Monitoring***

I am pleased to inform you that this has been approved.

Kind regards



Paula Simson  
Secretary to Faculty Research Ethics Committee

Cc. Dr Bogdan Ghita

Faculty of Science and Engineering T +44 (0) 1752 584 584  
Plymouth University F +44 (0) 1752 584 540  
Drake Circus W www.plymouth.ac.uk  
PL4 8AA

Mrs Jayne Breen  
Head of Faculty Operations