Device-type Profiling using Packet Inter-Arrival Time for Network Access Control

Musa Abubakar Muhammad

Cyber Technology Institute
De Montfort University, Leicester

This thesis is submitted to De Montfort University in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

June 2021

Dedication

To my Parents

For all their love, prayers and patience; without their endless support and prayers; I would not have accomplish this research. Though my **DAD** is not alive whom will be proud of this great achievement. I always pray to Almighty Allah to grant them the highest place in Jannah (Ameen).

To Usaini Garba Alhaji

For his endless financial support, advice and encouragement, from the beginning of my Undergraduate Studies to this stage, I am forever indebted. May Almighty Allah reward you with whatever good you desire.

To my Wife

For her endless love, support, and patience. Without her patience, most of this work would not have been completed.

To my Daughter and More to come

Thanks for coming into my life to complete this part of my studies together, I am proud of you. Also, thanks to those coming into the next chapter of my life, i will be proud of you as well.

To My siblings

Thanks for your prayers and advice, I am forever grateful

Declaration

I declare that the work described in this thesis is original work undertaken by me for the degree of Doctor of Philosophy, at Cyber Technology Institute at De Montfort University, Leicester, United Kingdom. No part of the material described in this thesis has been submitted for any award of any other degree or qualification in this or any other university or college of advanced education.

Musa Abubakar Muhammad
June 2021

Acknowledgements

All thanks to Almighty **Allah** Subhanahu Wata'ala for all the blessings and guidance bestowed upon me throughout the process of completing this research work. I would also like to extend my gratitude to several people for their undeterred support.

I would like to thank my supervisors **Professor Aladdin Ayesh** and **Dr. Isabel Wagner**, for their support and guidance throughout the stages of my research work at De Montfort University. Without their counsel, this thesis would not have reached this level.

My profound gratitude goes to my lovely **wife** who stood by me, advised, encouraged, supported, and prayed for me throughout my research journey, I love you baby. Also, special gratitude goes to **my daughter** (the joy of my life) who is always with me when I am back home, crying, playing and always making me happy with her lovely smiles. My thanks also goes to my parents, siblings, and in-laws for their utmost support and reassurances.

My sincere appreciation to my colleagues, friends, well wishers and relatives who have been praying and advising me on how to move forward throughout the period of my studies. Finally, special appreciation goes to the computer science and informatics department's PhD students and Lecturers at De Montfort University, especially to those that I can always count on for support during difficult times. I am forever grateful.

Abstract

Network Access Control (NAC) systems are technologies and defined policies typically established to control the access of devices attempting to connect to enterprise networks. However, NAC limitations have led to security threats that can lead to illegal and unauthorised access to networks as well as insider misuse. Current NAC configuration settings rely on point of entry authentication systems including passwords, biometrics, two-factor, and multi-factor authentication to protect employees, but this reliance can lead to security susceptibilities that can significantly damage enterprise network systems. In addition, incorporating NAC into the growing Bring Your Own Device (BYOD) paradigm further increases the security threats, vulnerabilities and risks potentials in enterprise network environments. Regardless of any existing security solutions, such as anti-malware, anti-virus and intrusion detection and prevention systems, security issues continue to rise within BYOD, with a proportionate increase in consequences and impacts.

This thesis explores novel solution paths to the above challenges by investigating device-type fingerprinting and behaviour profiling to improve the security of NAC. This is achieved by proposing a novel Intelligent Filtering Technique (IFT) that uses packet Inter-Arrival Time (IAT) data for smartphones, tablets and laptops to profile and identify abnormal patterns based on device-types. The IFT is composed of three data mining algorithms, namely K-means clustering, clustering-based multivariate gaussian outlier score, and long short-term memory networks algorithms. These algorithms are capable of identifying abnormal inter-arrival time patterns based on device-types. Despite the complexity of these algorithms and the huge volume of datasets involved, the IFT produces good results with high identification accuracy and a low number of false positives.

The effectiveness of the proposed technique is evaluated using a combination of datasets from different network traffic protocols, such as Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Internet Control Message Protocol (ICMP), as well as synthetic datasets. The results of the evaluation indicate good performance, with accuracies above 99%, and show that the IFT can be generalised. To the best of

the author's knowledge, this is the only technique to date that can identify abnormal inter-arrival time patterns based on the device-type. The new technique can improve intrusion detection system capabilities and outcomes by using device-type profiling to reduce the false positive rates of detected abnormal patterns.

Keywords:

Bring Your Own Device, Network Access Control, Device-Type Fingerprinting, Behaviour Profiling, Intelligent Filtering Technique, Outlier Detection, Long Short-Term Memory Networks

Publications

1. Musa Abubakar Muhammad, Pooneh Bagheri Zadeh, and Aladdin Ayesh. Improving security in bring your own device (byod) environment by controlling access. In *Faculty of Technology Conference*, pages 1–4, Leicester, UK, 2017. DMU, Dora

2. Musa Abubakar Muhammad, Aladdin Ayesh, and Pooneh Bagheri Zadeh. Developing an intelligent filtering technique for bring your own device network access control. In *Proceedings of the first International Conference on Future Networks and Distributed Systems*, ICFNDS '17, pages 1–8, Cambridge, UK, 2017. ACM

3. Musa Abubakar Muhammad and Aladdin Ayesh. A behaviour profiling based technique for network access control systems. *International Journal of Cyber-Security and Digital Forensics*, 8(1):23–30, 2019

4. Musa Abubakar Muhammad, Aladdin Ayesh, and Isabel Wagner. Behavior-Based Outlier Detection for Network Access Control Systems. In *Proceedings of the 3rd International Conference on Future Networks and Distributed Systems*, ICFNDS '19, pages 1–6, Paris, France, 2019. ACM. ISBN 978-1-4503-7163-6. doi: 10.1145/3341325.3342004

Table of contents

List of figures

List of tables

List of Abbreviations

Acronyms / Abbreviations	
AI	Artificial Intelligence
API	Application Program Interface
AUC	Area Under the Curve
BiLSTM	Bidirectional Long Short Term Memory
BN	Bayesian Networks
BYOD	Bring Your Own Device
C0	Cluster 0
C1	Cluster 1
CAIDA	Center for Applied Internet Data Analysis
CBLOF	Cluster-Based Local Outlier Factor
CMGOS	Clustering-based Multivariate Gaussian Outlier Score
CRAWDAD	Community Resource for Archiving Wireless Data At Dartmouth
CSV	Comma Separated Value
Db_index	Davies Bouldin Index
DM	Data Mining
EAP	Extensible Authentication Protocol
EMM	Enterprise Mobility Management

FN	False Negative
FP	False Positive
FPR	False Positive Rate
IAT	Inter Arrival Time
IDS	Intrusion Detection Systems
IEEE	Institute of Electrical and Electronics Engineers
IFT	Intelligent Filtering Technique
IoT	Internet of Things
LDCOF	Local Density Cluster-Based Outlier Factor
LSTM	Long Short Memory
MacOS	Mac Operating System
MAM	Mobile Application Management
Max	Maximum
Mbps	Mega bytes per second
MCD	Minimum Covariance or Determinant
MDM	Mobile Device Management
MIM	Mobile Information Management
Min	Minimum
ML	Machine Learning
MM	Markov Models
MSE	Mean Squared Error
NAC	Network Access Control
NAR	Nonlinear Auto-Regressive
NARX	Nonlinear Auto-regressive with External Input

NN	Neural Networks
NPV	Negative Predictive Value
ODDS	Outlier Detection Datasets
OS	Operating Systems
PPV	Positive Predictive Value
Q1	Quartile 1
Q2	Quartile 2
Q3	Quartile 3
R	Regression
RADIUS	Remote Authentication Dial-In User Service
ROC	Receiver Operating Characteristic
RV	Random Variable
SIEM	Security Information and Event Management
SMS	Short Messaging Service
SPC	Specificity
Std	Standard Deviation
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate

Chapter 1

Introduction and Overview

The incorporation of network access control into Enterprise Mobility Management (EMM) solutions still does not leave BYOD enterprise network without security limitations. This thesis attempts to address some of the network access control limitations in relations to the BYOD-type EMM solutions and their enabling technologies. Specifically, section 1.1 describes various enterprise mobility management solutions and network access control systems. A detailed background of the study is then presented in section 1.2 to provide understanding on NAC systems and their associated limitations. The research aim, objectives, and research questions are presented in sections 1.3 and 1.4. Section 1.5 details the main contributions made by this research, and section 1.6 summarise the chapter and present the overall thesis structure.

1.1 Introduction

BYOD is a policy that allows employees to bring the devices of their choice to access the enterprise resources that allow them to perform their daily work-related tasks. The widespread adoption of BYOD lead to a technological innovation in an enterprise network with policies that deliver new and significant business capabilities. The most commonly used personal devices for BYOD are smartphones, tablets and laptops [5], [6], [7], with the enabling technologies for BYOD being EMM solutions and Network Access Control systems [8], [9]. EMM is a set of solutions, processes and technologies designed to manage, monitor and control mobile devices in the workplace [10]. The basic function of EMM is to work in conjunction with NAC to prevent unauthorised and illegal access to enterprise networks [11], [12].

EMM involves a combination of three technologies, namely Mobile Device Management (MDM), Mobile Application Management (MAM) and Mobile Information

Management (MIM), to monitor and control devices in an enterprise network environment. MDM is security software that is implemented using an Application Program Interface (API) to manage, monitor and control the data on mobile devices [13]. It can be deployed on multiple platforms and supports multiple Operating Systems (OSs) to provide additional security measures [8]. MDM enforces security policies on a device to allow for the possibility of encrypting and wiping data both locally and remotely. The security policies are enforced to authenticate and install digital certificates as well as create individual profiles for each device connecting to the BYOD platform. Also, it has Over-the-Air (OTA) mechanism that helps to deliver the initial MDM configurations to install and remove applications, lock/wipe a device, remote back-ups of data and restoration of files [13]. MDM is intended to optimise the security and functionality of mobile communication networks while minimising costs and downtime. Traditionally, MDM supports the installation of an enterprise application on a device called an 'agent'. An agent is used to communicate with the enterprise management servers to transfer users' data and apply the relevant security policies on the device in question [14]. MAM and MIM are added to enterprise mobility management solutions due to MDM limitations and inability to separate between the personal and corporate data spaces. MAM is software responsible for provisioning and controlling access to mobile applications. MIM emerges as an add-on to maintain the integrity of enterprise information by encrypting data in a secure container in a remote location and share them between different endpoints and platforms [14], [15].

NAC systems are technologies and defined policies aimed at controlling network access to devices attempting to gain access to enterprise networks. An enterprise network is a complex and dynamic environment that includes the communication backbone for computers and other devices across different departments in organisations and workgroup networks accessing and sharing data [16]. The purpose of NAC is to control access to the enterprise network in order to prevent unauthorised users and workgroups from gaining such access. The enterprise network integrates all operating systems including Windows, macOS, Unix, and related devices such as smartphones, tablets and laptops to access data through client-server applications that allow for user authentication. NAC systems unify endpoint security solutions to manage and prevent unauthorised or illegal devices from accessing enterprise networks. The endpoint security solutions use a set of protocols to define the policies that secure devices in the initial network access stage [17]. These policies are defined to ensure that devices are correctly configured and operating efficiently. NAC has functions that work with EMM to verify protection against viruses, botnets, malware, spyware and other security

threats that can be transmitted across networks [18]. Another function of NAC is to ensure that devices are compliant with enterprise policy requirements. These policies ensure that before the devices are allowed access to the associated enterprise network, they are checked against security patches and updates, among others [19]. NAC has gained increasing popularity due to the tremendous growth in BYOD and the fact that enterprise networks no longer defend on the traditional security measures [20]. NAC security standards operate in IEEE 802.1x protocols to define and encapsulate network traffic based on the Extensible Authentication Protocol (EAP) over IEEE 802 [21]. These are established standards based on traffic encryption and integrity that protects enterprise networks from unauthorised and illegal access.

Incorporating NAC functions and capabilities into EMM solutions still does not leave BYOD enterprise networks completely immune and without problems. In particular, NAC in BYOD-type enterprise networks still have security vulnerabilities that can be identified and exploited by attackers. Ideally, BYOD enterprise networks would have no or significantly reduced problems with the application of NAC technologies. However, as information technologies advance to resolve certain problems, new ones inadvertently emerge, including the security context. It is only rational to engage in continuous efforts to address problems as they emerge and are recognised. Hence, this study considers some of the problems in improving NAC in BYOD enterprise networks.

1.2 Background of the Problem

As smartphones are becoming more powerful in workplaces, many organisations are enjoying the convenience of BYOD as it reduces the cost of buying and maintaining their own devices [22]. As well as improving the productivity of those employees using such devices, they are more comfortable using them anytime and anywhere [23]. Many organisations consider BYOD to represent an opportunity rather than a challenge [24], [25], [26]. Although employees can have access to enterprise networks with their devices at any time and anywhere, this can give rise to many significant challenges [27]. Most of these are related to access control, in which an attacker uses tricks to bypass an organisation's security to access their network and steal valuable data (illegally). In order to protect organisations against security attacks and network access challenges, the organisations themselves should be the key enablers for all the services they provide to their employees. These services can include network access, infrastructure upgrades, phone bills, training and support, among others, else the security issues remain the same. However, the proliferation of BYOD has resulted in increased complexity of

enterprise network security as a result of inconsistent policies and management controls across both their wired and wireless segments [12], [28]. Other concerns may arise due to lack of sufficiently innovative solutions to control the employee devices accessing the enterprise networks [29]. Such insecurity renders mobile devices open to malicious attackers. Once an attacker possesses a lost or stolen device, they can become, in effect, an internal user and use the sensitive information contained on the device to cause considerable damage to the organisation's network [30]. Once this situation occurs, all the network access control systems employed by the organisation would be rendered ineffective [15], [31]. As security is a key concern for BYOD platforms [32] due to the susceptibility of mobile devices to malicious attacks [14], current security measures need to be strengthened to overcome the security challenges posed by BYOD and to sustain its benefits to organisations. For example, where a device and/or access credential that belongs to valid user is used by an unauthorised user to access the enterprise network/platform, traditional access control systems are not able to detect such impersonation and unauthorised access given their reliance on correct devices usage and/or user credentials. Enterprise networks should not depend or rely on traditional security measures such as; antivirus and anti-malware solutions, intrusion detection and prevention systems, among others, to overcome the above challenges [33]. Although these security measures help to mitigate unauthorised security intrusion, however, scanning mobile devices can quickly drain their batteries. Mobile devices bear limited power that typically lasts less than eighteen hours. Thus, controlling access from the network would be a better solution approach by which to deal with the above BYOD security challenges [34]. For example, having a post-authorisation system that would consider the behaviour of logged-in devices from the traffic packets they generate to help differentiate between normal and abnormal device-types, and control access to network resources. This can help to resolve the issue of recognising abnormal devices to control access while not wearing out device power.

NAC systems use a set of protocols to define and implement security policies that provide secure network access to authorised devices during the initial network access stage. These policies are used to authenticate and authorise devices that comply with the enterprise's predefined policies. NAC has many advantages such as visibility control to notify network administrators about policy violations, manage and inspect configured network devices, and enforce access control policies to the devices connecting to enterprise networks, among others [35]. NAC has various limitations, however, such as the inability to detect advanced persistent threats, and weaknesses in identifying devices since defined policies might only allow users to pass their credentials to the

Remote Authentication Dial-In User Service (RADIUS) server but are not applied to devices [36], [37]. These limitations could lead to security attacks where an attacker can steal users' credentials and access an enterprise network with a different device to flood the network with packets. The attacker can also use their personal devices to spread ransomware infections to all the devices across the network. All these potential issues create unique challenges that can seriously affect enterprise networks because mobile devices are increasingly being targeted by criminals [31]. An intelligent filtering technique (IFT) can address these concerns by looking into the variation of packet inter-arrival time patterns of the devices and device-types connected to enterprise networks to identify and filter abnormal network traffic patterns or abnormal devices.

1.3 Aim and Objectives

The main aim of this research is to develop an intelligent filtering technique with the capability to fulfil the requirements for more effective network access control security for BYOD systems. The technique will be implemented according to three steps. The first of these focusses on the identification and analysis of a dataset suitable for the research. The second step focusses on developing a device-type profiling approach using the unlabelled dataset analysed. The final step focusses on the use of the labelled dataset from the device-type profile to develop and implement the IFT. In order to achieve the above research aim, the following research objectives will be followed:

1. To investigate the current cybersecurity threats, vulnerabilities, attacks, and security requirements for BYOD-based Network Access Control systems, and to understand the feasibility of detecting and filtering abnormal network traffic patterns.

2. To investigate the application of Artificial Intelligence (AI) techniques and appropriate datasets for achieving device-type post-authorisations network access control security in BYOD enterprise network.

3. To explore and analyse the datasets identified in (2) using an applicable AI technique to gain appropriate insights and support the validity of the device-type profiling approach.

4. To develop a behavioural profiling approach for classifying and labelling the network packet inter-arrival times to help with the identification of abnormal device(s) or device-types that suggest security threats.

5. To develop an intelligent filtering technique for a BYOD enterprise network based on device-type profiles, and validate its effectiveness in different experimental scenarios using appropriate performance metrics.

1.4 Research Questions

This research formed certain criteria based on the research aim and objectives to judge whether and to ensure that the research has answered the following questions:

1. What are the security risks (threats, vulnerabilities, attacks), security requirements associated with BYOD-enabling technologies, and the available security measures in the domain for addressing the risks and requirements?

2. What are the applicable outlier detection techniques used for addressing network access control problems and requirements, their strengths and limitations, and available datasets that fulfil BYOD requirements?

3. Can device-type profiling using packet inter-arrival time be useful for identifying abnormal devices in BYOD networks/platforms?

4. Can intelligent filtering technique that is based on device-type profiling support better identification of abnormal device-types in BYOD enterprise networks?

The above research questions will be revisited in the conclusions to ensure that they have been successfully answered and proven (or otherwise) experimentally.

1.5 Contribution

This research contributes to the field of NAC by developing a device-type IFT intended for the automatic identification of abnormal device-types that suggest threats using packet inter-arrival time patterns. The security technique is novel as it uses a post-authorisation approach with machine learning clustering and classification techniques to differentiate abnormal network traffic packets from normal ones and thereby control network access. Furthermore, a unique inter-arrival time data analysis and classification technique have been developed to explore datasets and validate the device-type profiling assumptions.

The proposed research experimental settings are unique and present an improvement of known concepts available in this area as current works use clustering techniques to

assume normal and abnormal patterns (chapter 3). In addition, other researchers draw on classification techniques, e.g. neural networks, to classify normal and abnormal patterns (chapter 3). As an improvement, the technique uses K-means clustering to understand the inter-arrival time patterns of the data from the devices (chapter 4), and then uses clustering-based multivariate gaussian outlier score (CMGOS) to distinguish and label normal and abnormal inter-arrival time patterns (chapter 5) and the long short-term memory (LSTM) to train the IFT to identify abnormal inter-arrival time patterns (chapter 6). The novelty of this approach also involves a series of experimental demonstrations using inter-arrival time datasets and synthetic data to evaluate the effectiveness of the technique (chapter 7).

1.6 Chapter Summary

In the above chapter (1) the need for an improved security technique for detecting abnormal patterns in BYOD enterprise networks was introduced. BYOD and its enabling technologies were also introduced in addition to discussing the current problems around the inherent security issues and risks in the enabling technologies of BYOD enterprise networks. Then, the main aim of the research was set out, which is the development of device-type intelligent filtering technique. The aim was explored following the set of objectives and research questions are described in sections 1.3 and 1.4, respectively. The main aim and objectives and research questions are addressed in all the chapters presented in the thesis. Then, the main contribution of this research is presented. The remaining chapters can be outlined as follows:

Chapter 2 presents a literature review of the network access control domain. The chapter starts by introducing NAC systems, and their background, history and features. It also reviews enterprise network security with a view to determining and identifying the security requirements for BYOD. It then investigates the most prevalent security attacks instigated against NACs and the available security solutions with a view to identifying their limitations and ways to improve them.

Chapter 3 describes and justifies all the AI techniques and algorithms used in this research. The chapter starts with a background study of machine learning and data mining, and then discusses all the algorithms used in this research along with the justifications for doing so.

Chapter 4 describes the datasets available, the justifications for their uses and gives detailed descriptions of them. The datasets described contain the packet IAT traffic of three different networks measured via active, isolated and passive network monitors.

Chapter 5 conducts an experimental investigation in order to label and classify the datasets into normal and abnormal profiles. The chapter describes a device-type profiling, appropriate experimental settings, and analyses the experiment results gained from active, isolated and passive network traffic datasets. The last section of the chapter labels the data for each device-type into normal and abnormal based on patterns identified by the outlier detection algorithm, specifically clustering-based multivariate gaussian outlier score.

Chapter 6 develops an IFT based on device-type; it starts by describing the IFT implementation using the Long Short Term Memory (LSTM) algorithm, and then conducts experiments using the classified and labelled active, isolated and passive network traffic datasets based on device-type, analyses the results, and evaluates the performance of the IFT.

Chapter 7 evaluates the performance of device-type IFT based on the original and synthetic datasets. It describes and demonstrates synthetic data generation from active, isolated and passive network traffic datasets, and then evaluates the results in two distinctive scenarios (such as network traffic rate comparison and synthetic data generated from the original datasets) to demonstrate the effectiveness of IFT for each device-type from active, isolated and passive and network traffic datasets.

Chapter 8 gives a series of concluding remarks to the thesis, highlighting the achievements, and possible future research in this area.

Chapter 2

Network Access Control Systems

Based on the findings from the literature review, a more specific introduction to NAC systems is given. Briefly, section 2.1 details their associated background and features and presents the prevalent security challenges and ways to address these, including currently available security solutions. A background study is then provided in section 2.2 for enterprise security solutions to NAC systems by focusing on research that has the potential to provide adequate security for enterprise networks. Subsequently, section 2.3 presents the previous works that use packet inter-arrival times to enhance NAC security based on profiling and fingerprinting approaches. Finally, the chapter is summarised in section 2.4.

2.1 Network Access Control systems (NACs)

NAC unifies endpoint security solutions to enable access control and enforce security policies on devices connected to an enterprise network. NAC policies offer the capability to identify devices connected to enterprise networks and restrict those that do not comply with the organisation's policies [38]. Hence, devices must be policy-compliant to be allowed access to the appropriate enterprise resources. These policies include security patches, firewalls, and anti-virus and anti-malware updates. NAC consists of pre-admission and post-admission phases based on whether the policies are enforced before or after the devices gain access to the enterprise network. In the pre-admission phase, the devices are inspected prior to being allowed to access the enterprise network, whilst the post-admission makes enforcement decisions based on employee actions after being granted access. NAC includes functions that block infected devices from spreading malicious code across the enterprise network [39].

NAC enables network administrators to configure devices to access the enterprise network through wired and wireless access points. NAC has a powerful authentication mechanism that verifies the connected devices through the wired and wireless access points. Additionally, NAC depends on Local Area Network (LAN), Wide Area Network (WAN) and Virtual Private Network (VPN) remote access to protect organisations from security threats and attacks. It also enforces access control policies to ensure that trusted devices can access the network. The increasing prevalence of BYOD is the key reason why NAC is increasingly becoming an in-demand technology [29], [40].

2.1.1 Background of NAC

Network access control emerged in 2006 to block unauthorised devices from traditional data centre networks. As the technology evolved, stronger NAC standards evolved due to the attention from EMM vendors, such as Microsoft, Cisco, the Trusted Computing Group, and Samsung Knox [41], [42], [43], to meet the challenge of applying security policies that work across BYOD networks [10], [31]. Over the past decade, NAC solutions have proved difficult to implement without cross-vendor integration. Vendors integrate EMM solutions with traditional security measures, such as anti-virus, anti-malware and intrusion detection systems, to protect enterprise networks. This integration makes NAC more robust as it hinders network access and filters devices that do not comply with enterprise policies. However, it has evolved to scan and block PCs and endpoints that are not registered on the network, however, it has evolved to authenticate and authorise those devices that comply with corporate network policies. As organisations are rapidly adopting BYOD, they must ensure that attackers have not compromised enterprise network access. Before the adoption of NAC, enterprise networks were self-contained within a well-defined perimeter to prevent attacks. Currently, this simple perimeter no longer exists due to the addition of mobile and BYOD devices to enterprise networks. These enterprise networks are accessed by a great range of endpoints from a large number of locations. Therefore, enterprises must ensure that the devices connecting to their networks adhere to their security policies and that the policies support multiple non-standard devices per user.

The IDC report [44], [45] predicted that the NAC market would grow by 31.17% with a global revenue of $7.065 billion by 2020. Similarly, Grand View Research [46] projected a 30.2% compound annual growth rate by 2022, with the industry reaching $4.39 billion in annual revenue by then. The IDC report [44] also forecasted that BYOD would continue to grow from 96.2 million mobile workers in 2015 to 105.4 million by 2020. The explosive growth of these endpoints creates an expanding perimeter that must

be contained. This tremendous increase in the use of mobile devices has contributed to our modern-day life [47], in which ubiquitous sensors are enabled by Wireless Sensor Network technologies [48]. Therefore, current NAC solutions are capable of registering, profiling and blacklisting devices that do not comply with organisations' predefined policies. NAC can be used with a variety of enabling wireless technologies, such as radio-frequency identification tags, embedded sensors and actuators [49].

2.1.2 Related Works in NACs

Several studies have been conducted to improve NAC solutions, some of which are based on open-source NAC solutions, mobile virtualisation and virtual private network approaches. For example, open-source solutions are freely available for organisations to modify according to the organisational requirements. A mobile virtual machine uses virtual mobile infrastructures to run a mobile application and OS on a remote server to effectively redefine EMM, thereby securing and supporting multiple devices without wrapping or the modification of the underlying mobile device OS. It also provides flexibility for users by separating their personal data from enterprise data based on virtual machine configurations. The data are encrypted using different keys to ensure that other users or network administrators cannot access the data stored by the user. Mobile virtual private networks allow users to connect to the enterprise network through an encrypted tunnel.

Open-Source NAC Solutions

Lin et al. [50] proposed a flexible NAC solution that uses a virtual desktop environment to automatically enforce access control security policies for mobile devices. The solution is flexible to the extent that an administrator can configure multiple devices for a single user. The user can access enterprise resources through a virtual desktop environment using one device at a time. Another piece of research by Matias et al. [51] proposed a flow-based NAC using an extended version of IEEE 802.1x to provide an effective and secure authentication mechanism in the proactive mode. It provides simultaneous authentication, authorisation and the proactive enforcement of defined network traffic requested from specific services. These specific services are uniquely provided by the service providers to avoid network collisions.

Inverse Inc. [52] developed a fully trusted open-source NAC solution called PacketFence. PacketFence has a feature that enforces access control compliance policies along with remediation, registration and centralised access management on both wired

and wireless devices. It is integrated with the Snort/supricana intrusion detection system and a vulnerability scanner to support the 802.1x layer2 isolation of problematic devices. It also has an inbuilt intrusion detection system function that verifies and monitors devices after authentication. Chao [53] proposed another open-source solution called FreeNAC. The solution integrates 802.1x and Cisco virtual local area network Membership Policy Server port security to provide access control, switch management and live end-to-end network discovery. It has a function that can track devices on the network and control how they access network resources with visibility regarding their network usage. It also queries switches to alert network administrators to obtain more information about new devices and users connected to an enterprise network. Furthermore, other solutions available in the market, such as VMware, IBM, and Samsung Knox, can also improve NAC based on mobile-centric technologies [54].

Mobile Virtual Machine Approaches

The VMware Mobile virtualisation platform presented in [55] uses desktop virtualisation to deliver end-to-end solutions and facilitate the use of user-owned devices in the workplace. It separates personal and enterprise workspaces based on type 2 hypervisors. This solution is easy to use and avoids the need to install EMM solutions on personal mobile devices. It also allows users to run multiple operating systems simultaneously on the same device. The solution is coherent, although it cannot obtain the high capabilities needed to execute in privileged mode because it relies on certain trusted components. The limitation of this architecture is that VMware has a predefined environment, and employees and/or organisations cannot redefine this environment according to their preferences. Andrus et al. [56] proposed a lightweight virtualisation architecture that enables a single mobile device OS to run multiple virtual machines simultaneously. The architecture is configured to separate the virtual enterprise environment from the personal mobile environment to ensure that the device in question is secure. Generally, this architecture is efficient; it adopts kernel modification techniques and does not require the user to run multiple OS instances. It gives users the flexibility to run multiple mobile virtual phones on one device and display only a single application at a time. However, this framework is limited to Android devices and the virtual environment is separated, resulting in certain limitations regarding user interaction and giving rise to certain privacy issues.

The para-virtualisation approach proposed in [57] allows users to perform multiple tasks from different enterprise locations based on predefined security profiles. Users are required to connect their personal mobile devices to the enterprise network before they

are given access to the security profile. The security profile was configured to enforce software isolation policies on the Android platform, separating work data from device data at the same time. Each profile can be associated with one or more context, which determines when the profile is activated. However, while the framework described above is efficient, it is limited to Android OS and does not support another mobile OS. It is also resource-intensive for mobile devices and is not amenable to automatic policy enforcement. It is also unlikely to be able to access the virtual platform through a mobile device without modifying the employee data. Further limitations to these approaches are discussed in detail in [18], [58], to which the reader is referred for further information.

Mobile Virtual Private Network Approaches

Chunle et al. [59] proposed an enhancement of NAC using mobile virtual private networks. This is a technique that establishes secure network access and communication through network terminals. Also, it adopts the characteristics of authentication, encryption, key management and compression algorithms in traditional virtual private networks. Mobile virtual private network technologies have, to date, been considered an optimised and parallel reposition of traditional virtual private networks. This traditional virtual private network has made considerable progress in decreasing transmission delays, increasing throughput and reducing computational overheads [60]. This solution can be implemented to prevent employees from gaining unauthorised access to their organisation's data [61]. Several research efforts have been conducted to improve the security of mobile devices when connecting through virtual private network channels.

Uskov [62] developed a systematic approach that generates different mobile virtual private network solutions to enhance NAC The solution is effective, but it needs system administrators to analyse a variety of virtual network architectures, topologies and/or a combination thereof to provide adequate access to enterprise networks. Also, Garg et al. [63] improved mobile device authentication through mobile virtual private networks based on the session key agreement approach. The session key provides users with secure authentication services to the Socks v5 protocol for a virtual private network using a mobile phone to access enterprise networks. The session keys are generated based on the mobile device's international mobile subscriber identity, which provides a unique identification. Another research effort by Chunle et al. [59] proposed the Communication Supportable Generic Model (CSGM) for mobile virtual private networks, which addresses the gaps between the mobile network environment and

mobile virtual private networks based on a scalable and stable virtual private network service.

Having the above security measures in place is efficient and can protect NAC systems from most security issues. However, the use of mobile devices to initiate connectivity to mobile virtual machines and virtual private networks can give rise to certain security concerns, particularly for wireless network connections. For this reason, additional solutions are required to tighten up NAC security issues. Also, due to the recent increase in the use of smartphones and tablets resulting from BYOD, reliable connections can be very difficult to achieve due to both human and non-human error [59]. The human factors involve users who are actively switching, opening or closing network access, whereas external environments are usually the cause of non-human errors. These factors can affect mobile virtual private network performance as well as cause frequent application failure, reduce productivity and increase data loss. Also, Information Security Officers may struggle to find the balance between data protection and implementing access control policies for their IT infrastructures [64]. These issues are critical when organisations do not own the devices used by their employees. Therefore, this research aims to develop solutions to secure NAC by looking into existing NAC features and finding ways to improve them.

2.1.3 Features of NAC

NAC system features are designed to enforce access control policies and mitigate against insider attacks within enterprise networks based on two-tier strategies implemented before and after access has been gained. These two-tier strategies constitute a pre-admission and a post-admission phase. The pre-admission phase is used to verify that the device attempting to connect to the enterprise network complies with an associated predefined set of policies, which include malware and antivirus detection and security patches installed on devices. If a device passes verification, it is granted access; otherwise, it is quarantined or blocked from the enterprise network. The post-admission phase controls the network hosts to ensure that the devices have complied with the appropriate enterprise policies. This includes monitoring the network traffic to detect deviations from normal network patterns and pushing security updates [65], [66], [67].

NAC uses the features described above to allow access control decisions based on intelligent agents. The intelligent agent can be installed or configured independently within a network to inform the network administrators about the behaviour of the end systems. This approach has been adopted by modern operating systems to provide network access protection agents as part of their releases [68], [69]. The agents work in

two ways, namely the out-of-bound mode and inline mode. The former is configured in a central server such that the agents can control the switches to enforce access control policies and distribute the policies to the end systems. The latter is a client solution that acts as an internal firewall for access-layer networks and enforces access control policies. The remediation strategy feature allows network administrators to deny network access to or otherwise quarantine users who do not have up-to-date patches installed on their devices. The most commonly used features of remediation strategies are quarantine and captive portals. The quarantine feature is implemented based on a Virtual Local Area Network (VLAN) to restrict users with routed access to specific hosts and applications, whereas the captive portal intercepts the network access and redirects users to update their security patches.

2.1.4 NAC Security Challenges

The NAC systems authenticate and authorise user access to an enterprise network through the installation of software applications on employee devices that passes their login credentials to a RADIUS server. Given that NAC is a client application that authorises and passes user credentials to a RADIUS server for authentication, it becomes clear that NAC ultimately authorises users – not their devices. Therefore, an employee can take advantage of this limitation to transfer their credentials to other devices and thus gain unauthorised access to the enterprise network. Another limitation of NAC is that it supports a limited range of devices, that is, Windows and MacOS and deems other devices, such as printers and gaming systems, unmanageable [70]. Also, after NAC authenticates a user, it lacks the functions to detect and deal with abnormal network traffic patterns [71]. Additionally, NAC lacks device authentication and management functions as it only authenticates users, not devices [43], [72]. These limitations can lead to various security challenges.

Specifically, these challenges include an attacker with an employee device stealing their user credentials and thus gaining access to the enterprise network through the device to, for instance, deceive the enterprise server by flooding the network with packets or to steal sensitive information [73]. Also, the attacker can use the limitations of NAC to cause damage to an organisation's reputation. Insiders can pose another kind of challenge if they are aware of these limitations; for example, as they cannot be detected, they can use their privileges to cause damage to an enterprise network. Perhaps one of the most significant forms of such damage is exposing customer and employee personal data [74]. Such breaches also include identity theft, the inappropriate use of data or the sale of sensitive information, leaving an organisation liable for the associated damage

and potentially leading to regulatory compliance. Also, a competitive organisation's position might suffer due to the trading of intellectual property for unauthorised purposes by internal employees [75], [76]. Other challenges lead to system damage, such as server downtime, infected enterprise devices, and disrupted business operations. Therefore, NAC systems should cover the limitations mentioned above to deal with the inherent security challenges affecting the implementation of BYOD in the majority of enterprise networks.

2.2 Enterprise Security Requirements for NAC

The security requirement is vital to the solutions to the different security challenges of enterprise networks. If critical security requirements are provided, then security will be much easier to achieve. Additionally, these security requirements (confidentiality, integrity, availability and non-repudiation) should be implemented in enterprise networks to provide high-level security, although it is impossible to implement all of them in a BYOD environment. They are designed to ensure that organisations can identify the set of key requirements they need to secure their systems. For example, this research aims to improve NAC systems, which are one of the key security requirements. Additionally, all security requirements designed by an organisation should achieve certain implementation objectives [77], [78], [79].

2.2.1 Overview of Security Requirements

- **Confidentiality:** The design of enterprise network security should ensure that only authorised users have access to authorised network resources; moreover, it should safeguard the privacy of the authorised enterprise resources from unauthorised users and attackers. Confidentiality can be obtained using any of the well-known mechanisms, such as encryption, authentication and access control. Authentication and access control prevent unauthorised users and attackers from accessing the transmitted information that passes through a network.

- **Integrity:** This ensures that the enterprise network guarantees that the data are not modified, deleted, removed, recorded, corrupted or retransmitted by unauthorised users, either intentionally or unintentionally. It must ensure that illegal, unauthorised users and attackers do not modify any data or information transmitted through the network, and there should also be an indication of

information replay, which is essential in daily operations where such changes could cause severe damage.

- **Availability:** This must ensure and guarantee that only legitimate devices have on-demand access to enterprise resources. Also, security systems should ensure that illegitimate users and attackers do not block access to wireless network resources because attackers can compromise the availability of a network and there is a need for mechanisms to safeguard such availability.

- **Non-repudiation:** This ensures that the receiver and sender cannot deny the reception or sending of data to or from other end devices. This approach can detect and isolate compromised end nodes. If Device A receives an erroneous message from Device B to break down Device A's security, Device A can thereafter 'accuse' Device B of sending erroneous information and expose Device B to other end devices to convince them that end device B is malicious and should not be routed through in the future. This is very important in cases of disagreement of this sort and can be obtained using digital signatures that relate the data, for example, fingerprinting techniques.

2.2.2 Overview of NAC Security Attacks

Enterprise networks have moved from wired to wireless technologies, which has had a negative impact on their security infrastructures. Generally, wired networks are easier to secure, and poor implementation leads to security vulnerabilities that can cause damage to an organisation. There has been significant failure or negligence in addressing the vulnerabilities identified in organisations. The most common security attacks in NAC systems are malware, eavesdropping, man-in-the-middle attacks, and advanced persistent threats [3], [80]

- **Malware:** This is software created with the purpose of damaging or disrupting the normal operation of applications and devices. It comes in the forms of viruses, spyware, worms and Trojan horses and is intended to gather information about other devices without permission. If there are insufficient or inappropriate malware prevention measures in place, they can cause great damage to the organisation, including information theft, interruption of business processes, the capture of valuable or classified information, and the deletion of valuable or sensitive data [14]. In terms of access control, this can lead to the denial of legitimate users gaining access to enterprise network resources, for example by

crashing the operating system, deleting the installed applications, formatting the device storage, draining the battery or massively increasing the central processing unit load [81]. Malware is one of the major contributors to cyber attacks in most organisations, especially when employees' own devices are allowed to access enterprise network resources [82]. Therefore, an organisation needs to ensure that the spread of malware infections is prevented through the use of antivirus and anti-malware solutions [83].

- **Eavesdropping attack:** This type of attack can affect enterprise networks when an employee accesses enterprise resources from a public network. Attackers can easily join the network and use network monitoring tools to steal login and valuable data that pass through the network. Mobile devices are particularly vulnerable, and attackers can take advantage of stolen devices and non-updated devices to steal enterprise data [84]. Integrity and confidentiality of information are potentially compromised due to employee negligence. Chang et al. [25] pointed out that enterprise data can be easily intercepted through public Wi-Fi. Moreover, the integrity of the information and the confidentiality of data can be compromised. However, it is also difficult for employees to differentiate between reliable and compromised data, also for wireless networks. As discussed in [5], this issue can be resolved by encrypting or tunnelling communication through a virtual private network when using public Wi-Fi.

- **Advanced Persistent Threat(APT):** The purpose of this form of attack is stealing intellectual property using multiple attack vectors. This attack can be executed by extending and establishing a foothold within the organisation information technology infrastructures with the purpose of exfiltrating information to weaken the critical features of the organisation [85]. This type of attack is stealthy and covertly targets organisations over an extended period. It can cause severe damage to organisational infrastructure [86].

- **Man-in-the-middle (MITM):** This refers to an application-based attack which occurs as a result of a vulnerability, or vulnerabilities, left by an application developer during development, in which the attacker persistently changes the URL of the server, caching the URL and taking control of the behaviour of the application to create loopholes for multiple vulnerabilities. This type of attack occurs in various forms, such as snooping for sensitive or confidential data, password stealing, denial of service and advanced persistent threat. All mobile OSs are susceptible to this type of attack and malicious spyware or key loggers

can be installed on such devices to open loopholes to efficiently carry out the attack, leaving the employee unknowingly and permanently loading the data from the attacker's site while sniffing sensitive organisational data [84], [86].

2.3 Review of Security Solutions for NAC Systems

Several research efforts have been conducted to improve NAC security using fingerprinting techniques, behaviour profiling and intelligent filtering techniques. Fingerprinting techniques include the active and passive collection of a configuration of attributes from devices during a network 802.1x communication to accurately identify the connected devices on the network from the wireless access points and conduct reconnaissance against a potential target [87]. Behaviour profiling has been applied in mobile devices to enhance security by monitoring unusual patterns or deviations from the normal behaviour of the network devices. It also involves the use of algorithms to find correlations in an enormous quantity of data, which can be used to identify the representation of the observed users, devices and applications to form an associated profile [88]. The intelligent filtering technique uses data from the behavioural characteristics of devices and users to block what appear to be abnormal patterns that deviate from normal network behaviour [89].

2.3.1 Fingerprinting Techniques

Research into fingerprinting began around 2010, focusing on querying web browsers to measure network traffic to identify contextual information about the devices in a network. The fingerprinting technique can also be used by attackers to conduct reconnaissance against a target. Device-type fingerprinting is an approach to extracting unique features from devices to generate device-specific signatures and use these to identify the device-type. It consists of two techniques, namely passive and active. The former relies on TCP/IP configurations, OS, clock-skew and wireless settings to gather contextual device information, and it does not involve querying a client machine to obtain the device fingerprint. The latter involves the installation of an application on a client machine to query the switches to gain access to device information, such as MAC address, serial number, IMEI number, etc. The existing fingerprinting techniques in the network access control domain is presented in Table 2.1.

Franklin et al. [90] developed a passive technique that fingerprints wireless device drivers running on 802.11 networks. The technique is unique in that an attacker

Table 2.1 A review of fingerprinting Techniques

Approach		Detection Type	Features Used
Franklin et al.	[90]	Device Driver Fingerprinting	Time delta
Desmond et al.	[91]	Device Fingerprinting	Time delta
Gao et al.	[92]	Access Point Fingerprinting	Inter-arrival time
Jana et al.	[93]	Device Driver Fingerprinting	Time delta
Neumann et al.	[94]	Device fingerprinting	Transmission time and IAT
Arackaparambil et al.	[95]	Fake access point Fingerprinting	Clock-skew
Corbett et al.	[96]	Vendor-type Fingerprinting	Rate Switching
Radhakrishnan et al.	[97]	Device and Device type Fingerprinting	Inter-arrival time
Xu et al.	[98]	Device Fingerprinting	Transmitted signal frames and location
Dalai et al.	[99]	Device type Fingerprinting	Probe request frames

can use it to conduct reconnaissance against a potential target without requiring physical access to modify the wireless drivers. This fingerprinting approach uses the active scan functions of wireless clients to probe request access points to identify the drivers employed on a device in a passive manner. The limitation of this approach is that it cannot identify or differentiate versions of the same driver and it cannot work if there is hardware abstract layer. Desmond et al. [91] proposed a passive fingerprinting technique that uniquely differentiates devices through a timing analysis of 802.11 probe request frames. The advantage of this technique is that it can be used for spoof detection, reconnaissance and the implementation of access control against masquerading attacks. This technique can be affected by network interference, shadowing congestion, and packet loss, which may reduce its efficiency.

Gao et al. [92] proposed a black box-based fingerprinting technique that fingerprints a network to identify the available access points. The approach can be used as an offensive or defensive measure, depending on the situation. For example, network administrators can use it to detect a rogue access point, whilst attackers may use it to fingerprint access points to launch firmware-specific attacks. The proposed fingerprinting technique by Jana et al. [93] uses a clock-skew method to fingerprint a network to detect unauthorised access points. The clock-skew can be estimated using the time synchronisation functions of a probe request-response and beacon frames to differentiate between authorised and fake access points. Neumann et al. [94] proposed a fingerprinting system that detects illegal and suspicious devices connected through 802.11 networks. The systems were designed to enable two similar devices to have distinct signatures based on call diversity and anti-forgery mechanisms. The call diversity slightly changes the network traffic attributes of the fingerprinted devices

to uniquely identify their signatures. The anti-forgery mechanism in this system is proposed to make it difficult for an attacker to forge a device fingerprint as well as prevent replay attacks and separate forged from legitimate signatures. The proposed wireless device fingerprinting approach by Arackaparambil et al. [95] uses a series of network packets to observe the responses of the target fingerprinted device; it then injects non-standard and malformed packets to distinguish between legitimate devices and non-legitimate devices that spoof the MAC addresses of legitimate ones.

The proposed fingerprinting approach by Corbett et al. [96] uses network interface cards from different vendors to develop a fingerprinting technique that identifies unauthorised access to a network. This approach fingerprints all the network interface cards and implements a spectral profile that can be used to separate legitimate and fake network interface cards from different vendors. The limitation of this approach is that it is limited to a number of devices, and increasing this number reduces its accuracy. The proposed work by Radhakrishnan et al. [97] uses packet inter-arrival times to fingerprint a series of devices and device-types to identify the variation of inter-arrival times for different devices and device-types. This approach relies on differentiating devices by looking into the statistical distribution of the packet inter-arrival time features generated by a given device for different applications while accessing a network. They assume that the physical features of devices, such as direct memory access controller, memory, processor(s) and clock skew, reveal how these devices transmit packets over the air. This can be used as an improvement to existing fingerprinting techniques by generating signatures from devices and device-types to exploit the heterogeneity of their functions based on different hardware configurations and variations in clock-skew.

Xu et al. [98] presented a tutorial on wireless device fingerprinting and discussed various features that can be used to generate device fingerprints and classify features based on protocol layers. Then, they analysed existing fingerprinting techniques and determined the best approach for fingerprinting mobile devices. They concluded that the best approach can be developed by combining fingerprinting approaches with localisation and tracking to mitigate attacks in wireless networks. Dalai et al. [99] proposed a technique that uses the homogeneity of wireless devices and measures wireless networks utilising a probe frame request to produce device-type signatures. The digital signature thus produced can be used to identify devices in a network. In contrast, our research work aims to improve the works related above based on profiling the packet inter-arrival time measurements of devices and their device-types to classify normal and abnormal patterns.

2.3.2 Behaviour Profiling

Behaviour profiling is a term commonly used in organisations to identify the working patterns of an individual employee. In this work, a behaviour profile is defined as a technique for detecting an abnormal pattern of devices in an enterprise network. Behaviour profiling has been applied in mobile devices to enhance security by monitoring for unusual patterns or deviations from normal network behaviour. It also involves the use of algorithms to discover any correlations in an otherwise enormous quantity of data. These data correlations can be used to identify the representation of the observed users, devices and applications to form a profile [88]. It is widely recognised that employees utilise their mobile devices in the workplace to be more productive. These mobile devices are used by employees to perform various tasks, for example, to access enterprise resources to perform their day-to-day work activities. Research into behaviour profiling is divided into two categories: mobile behaviour and network traffic profiling and are presented in Table 2.2. Mobile behaviour profiling focuses on contextual data, such as calling activity, mobility, and battery, to obtain a behaviour profile [100], whereas network profiling focuses on profiling the behaviour of network traffic activities [101].

Behaviour Profiling on Mobile Devices

Research into mobile behaviour profiling started in 1995, and it mainly focused on detecting abnormal patterns in mobile usage contextual data as obtained and monitored by network service providers. The research presented by Li et al. [105] proposed a behaviour profiling technique that continuously verifies mobile application usage based on a smoothing function, verification time and application nature. The proposed work was efficient as it was able to provide a continuous and non-intrusive verification of mobile users in three different modes, regardless of the device hardware. Also, the work was evaluated using a series of security scenarios in a simulated environment to demonstrate the effectiveness of the framework and to verify legitimate and illegitimate activities. The limitation of this work was that the MIT reality dataset [116] used in their approach only contains information suitable for behaviour profiling at that time. However, explosive growth in technology can have an impact on the use of this type of dataset, and in this case, the MIT reality data set was collected from 100 Nokia 6600 phones in 2004. The operating system they used is now significantly out of date and this device-type is longer manufactured, while the technology used in smartphones has been vastly updated.

Table 2.2 A review of behaviour profiling Techniques

Approach		Profiling Technique	Features Used
Giura et al.	[102]	SMS and Calling	SMS and voice call records
Saevanee et al.	[103]	Linguistic Profiling	Vocabulary and SMS style
Koh et al.	[39]	Usage Profiling	contextual data of mobile device usage
Kim and Kim	[71]	Service use pattern	device type, access time, access location and use time
Stoecklin et al.	[104]	mobile usage	users contextual data
Li et al.	[105]	Mobile usage	telephony, device usage and Bluetooth scanning
Kang et al.	[40]	Service use pattern	Access time, location and time of use
Hall et al.	[106]	Battery	service usage features
Buennemeyer et al.	[107]	Battery	battery run time
Shabtai et al.	[108]	Battery	device power, consumption, Bluetooth and Wi-Fi connection
Frias-Martinez et al.	[109]	Network	flows, average flow size, average flow duration, packets in all flows, average number of packets per flow, IP address and average packet size
Zhauniarovich et al.	[57]	Network users	users contextual data
Qin et al.	[110]	Network user	URL, IP address, DNS query logs
Tsompanidis	[111]	Movie streaming	Session, email synchronisation and web browsing
Chen et al.	[112]	Mobile traffic behaviour	location session, packet flow, user, type and duration
Jakalan et al.	[113]	IP network host	IP flow and time
Kihl et al.	[114]	User network traffic	Session length and traffic rate distribution
El attar et al.	[115]	Network users	CPU consumption, number of running process, memory usage and battery
Sun et al.	[31]	User mobility	Personal information, billing information, users home location and mobility

The behaviour profiling approach proposed by Giura et al. [102] uses contextual data from user sessions to profile the Short Messaging Service (SMS) and calling activities of network users. The profile runs in a background process to detect unusual call or SMS patterns for each user(s) and to prevent unauthorised access. The work was efficient, but it mainly focused on identifying abnormal user calling or SMS patterns. Saevanee et al. [103] presented a framework that applies secret-based knowledge and behavioural biometric techniques to provide non-intrusive, transparent and continuous authentication. The authentication is achieved without the knowledge of the user during device usage rather than when the device is switched on. The behavioural profile uses a linguistic profile for SMS for individual users based on their vocabulary and style. The attributes are extracted based on users' preferred words to form a behavioural profile. Thus, the framework manually selects user-abbreviated, emotional,

and favourite features as well as a set of linguistic features to automatically distinguish between users. The above-related work only focused on the implicit and transparent authentication of users rather than network access controls. Also, one of the challenges affecting the implicit and transparent authentication domain comprises the security and privacy threats posed to users' data [117].

Behaviour profiling based on network usage patterns has been proposed in various research efforts. Koh et al. [39] introduced a dynamic access control method using the contextual information of individual devices or the group to which they belong. The contextual data were collected from mobile device usage in agentless mode to develop a usage profile which is stored in a detection system. The detection system compares the security policies with abnormal behaviour to check for deviations and control access to the network. The behaviour-based anomaly detection technique proposed by Kim and Kim [71] can detect abnormal behaviour in the services used in a BYOD environment, while the profiles can be obtained based on classifying the network vulnerabilities occurring in this environment. Then, the contextual usage information of users (such as device-type, access time, access location and use time to pattern users access) can be used in the development of a usage profile, which can subsequently be used to detect abnormal user patterns.

Stoecklin et al. [104] proposed a technique that can be used to detect abnormal device(s) within an enterprise network. The framework introduced a non-intrusive big-data analytic method to obtain visibility regarding mobile device usage. The profiling data can be obtained directly from the devices' contextual data without the need to install an agent. Li et al. [105] developed a host-based multilevel behaviour profile for mobile intrusion detection. The profile is computed based on attributes such as telephony, device usage and Bluetooth scanning to form a behavioural profile for each device. The framework was efficient as it was able to detect different users' activities and protect devices against misuse through an application. Li et al. [118] developed a "sentinel", a behaviour profile approach that utilised application and service usage to verify individual users continuously to analyse their behaviour in real-time. The attributes used are telephone, text and web surfing, from which a behavioural profile is computed via a combination of the user's historical data with recent data to identify the legitimacy of their actions. A dynamic rule-based classifier is used to deal with the classification results accordingly.

Kang et al. [40] presented details of an abnormal behaviour detection method that used Bayesian theory. It was developed based on spam filtering in three stages, namely modelling the elements concerning the behaviour, patterning the behaviour,

and detecting abnormal patterns. The model considered the use of unstructured data in addition to network traffic. The attributes used were the type of device, access time, access location and time of use, thereby profiling the behaviour pattern. Then, those elements were classified using the probability of word occurrence behaviour to detect the abnormal activities in the device usage. Kim and Kim [71] proposed a behaviour-based anomaly detection technique that detected abnormal behaviour in the services used in a BYOD environment. The vulnerabilities were classified according to the user context. The attributes used in this instance were device-type, access time, access location and use time to obtain user's access pattern. The data were collected through the network traffic to analyse the service use behaviour and compare the past with the recent usage patterns. The above techniques only focus on usage profiling, and the excessive use of mobile devices in a network background can result in various security and privacy issues as well as increased energy consumption [119].

Various researchers have proposed mobility and battery profiling. Hall et al. [106] introduced an anomaly-based intrusion detection framework that examines the mobility of transport users. The behaviour profile was developed based on instance-based learning using calling, mobility patterns and service usage features. Buennemeyer et al. [107] proposed a battery-profiling system that alerted network administrators to deviations from normal behaviour. The work was developed based on an examination of smart battery drain times to ascertain the optimal transmission rate. The data used for profile observation were taken from nine device-types. The proposed work by Shabtai et al. [108] built unique profiles for smartphone batteries based on a network-centric environment. The profile monitored device power consumption and Bluetooth and Wi-Fi communication activity. The system was designed to continuously monitor the activities on the network such that when abnormal or attack activity was detected, the system alerted the network administrator. Also, it had other personalised functions that enabled the automatic disconnection of any given device upon an attack and the blacklisting of suspected activity until the profile was cleared of intrusion. This profiling technique suffered from various privacy issues, however, as it required the identification of users' locations and device batteries; also, this technique is somewhat outside the scope of this work. However, the review by Sharifi et al. [120] covered the intensive research on mobility and battery-profiling techniques to which the reader is referred for further information. In contrast to the abovementioned works, this research focuses on developing a behaviour profiling approach based on device-type.

Network Traffic Profiling

Research into network traffic profiling only recently began due to increased cyberattacks and the emergence of applications that can affect network traffic [121]. Network traffic models are aimed at detecting abnormal patterns and can be applied based on the selection of models that provide a good description of network traffic type and then estimate the parameters of the chosen model and test its accuracy to ensure suitability.

Frias-Martinez et al. [109] proposed a behaviour-based NAC based on the roles of each device host connected to a network using a group voting process. This process decides on the degree of similarity and makes appropriate decisions regarding the devices that can be allowed to access the network. The behaviour profiles are computed based on seven features (total number of flows, average flow size, average flow duration, the total number of packets contained in all flows, the average number of packets per flow, IP address and average packet size) from each host connected to the network. The cluster-based abnormal behaviour detection sensors are responsible for detecting abnormal behaviour in the profiles. An alert is sent only when a group of similar behaviours correspond to the host behaviour, and the profile raises an alert when the behaviour of the host does not correspond. Zhauniarovich et al. [57] introduced an integrated security system to resolve security threats by examining contextual information based on dynamic access control. The system collects users' contextual data in an agent-less mode following the authentication process. The contextual data thus collected are analysed to allow the detection system to determine abnormal behaviour in the process. It then forwards this to a control system to continuously monitor and control network access.

Qin et al. [110] developed a multilevel user cluster mining framework that measures user behaviour from different network prefix levels. Individual profiles are collected from network flow patterns, whereby the behavioural attributes that are extracted to compute the user profiles are URL, IP address, and DNS query log. The attributes are clustered and analysed to detect deviations from normal behaviour. Tsompanidis [111] proposed a behaviour-based traffic model that profiles movie streaming sessions, email synchronisation and web browsing. The profile is computed based on features such as session average bit rate and activity type to model the mutual interaction between the network performance and user-generated traffic. The model employs a Markov chain model to monitor behaviour. Chen et al. [112] proposed a model that characterises mobile traffic and the engaging behaviour of end-users. The features used for computing the profile are location session, packet flow, user type, and duration. The behaviour profile measures mobile device usage based on application interactions

and performance, and then compares this with previously determined volume metrics to detect intrusions. Jakalan et al. [113] proposed an IP network host behaviour profile that detects host dominant and persistent behaviour. The model uses IP flow and time to compute behaviour profiles and can easily detect an attack within a network upon deviation from normal behaviour.

Kihl et al. [114] analysed traffic measurements to model user behaviour, focusing on internet usage to identify users based on session length and traffic rate distribution. They also investigated user activities during a given period to form a usage profile for applications, user activities and traffic volume. Xie et al. [122] developed Bprofiler, which groups devices into intuitive groups within a hierarchical framework to profile individual user behaviour based on multiple dimensions. This technique does not require the installation of any application on the devices. Meanwhile, Kim [71] proposed a behaviour-based anomaly detection system that detects abnormal behaviour in the services used within a BYOD environment. A Bayesian network classifier is used to classify the probability of occurrence of a behaviour in this environment to compute behavioural patterns. The attributes used are device-type, access time, and access location and time to pattern users' access. This information is collected through network traffic to analyse the service use behaviour and compares past and recently used patterns to detect abnormal behaviour.

El attar et al. [115] proposed a behaviour-based detection system that relies on a light agent installed on the device to collect various access information, which is sent to a remote server using a secure socket layer connection. The attributes used for profiling are CPU consumption, number of running processes, memory usage and battery level. Sun et al. [31] proposed an efficient online abnormal detection algorithm using a data compression technique to identify user mobility, thus forming a behavioural pattern. The attributes utilised are users' personal information, billing information, and home location as well as identifying intrusions in users' movements. An alert is generated when the location is not registered on the system. In contrast to the works presented above, the current research aims to use packet inter-arrival time network traffic from device-types to identify abnormal network traffic patterns originating from the device-types.

2.3.3 Intelligent Filtering Techniques

Filtering techniques use data from the behavioural characteristics of devices and users to block abnormal network traffic patterns [89]. This can be accomplished using filtering rules, which are expressed using a set of training data or known facts about abnormal

network traffic. It enables network administrators to focus on attacks and apply filters to security event logs to extract access information and improve the discovery of frequent behaviour patterns. The filtering techniques are applied to mitigate the impact of attacks on the enterprise network.

The research work by Hajamydeen et al. [123] proposed a refined filter that can be used to retain the volume of abnormal network traffic in enterprise networks. The technique reduces processing time and improves anomaly detection by segregating network traffic into normal and abnormal. It then monitors these types of network traffic to check for abnormal patterns; if an abnormal pattern exists, the filter blocks it and allows normal patterns to continue flowing smoothly across the network. Jha et al. [124] proposed a filtering-based approach that regards normal traffic as noise and abnormal traffic as a signal. The technique sets a certain threshold on network traffic to estimate the strength of a 'signal' in a given network. If the signal strength is found to be above a specified threshold, this will indicate abnormal traffic and thus the trace will be flagged abnormal and blocked by the filter. Yu et al. [125] developed a filtering and refinement approach that detects abnormal behaviour in different mobile and web applications. The technique is applied to network traffic data that contain a small set of abnormal patterns. It separates the normal data instances and records them as normal patterns. Further, it generates an agent that is compared to the recorded patterns and that blocks any deviation so identified. A refinement was developed to improve the computational efficiency and effectiveness of the technique.

Lakhina et al. [126] proposed a subspace method of detecting abnormal behaviour in network traffic based on origin-to-destination flow to block unusual patterns. It uses a Kalman filter to extract the normal traffic patterns and principal component analysis to separate the normal from abnormal patterns. The identified abnormal patterns are then blocked by the filter. Knorn et al. [127] developed a simple predictive model that captures the baseline behaviour of central processing units. It partitions the behaviour into normal and abnormal patterns and blocks the patterns that deviate from the rest of the data. Handra et al. [128] proposed a filtering and refinement approach for more effective and efficient anomaly detection using DBSCAN. This method clusters all the instances and considers the abnormal clusters to be noise. Agarwal and Mittal [129] proposed a hybrid approach combining entropy and a Support Vector Machine (SVM) to detect abnormal behaviour in network traffic. This is achieved through calculating the entropy values of different network traffic features and training the SVM to classify the normal and abnormal traffic patterns.

Huang et al. [130] proposed an approach to filtering normal from abnormal behaviour in non-signature-based network traffic using principal component analysis and sketch-based and signal analysis. The approach compares past with the most recent patterns to detect abnormal behaviour. De la Hoz et al. [131] introduced a technique that uses a classification and self-organising map to detect abnormal patterns in the network. Principal component analysis and Fisher dominant ratio are used in feature selection and noise removal. The probabilistic self-organising map is also used to model the feature space and enable anomaly detection. Santos et al. [132] proposed a spam filtering technique for anomaly detection, which reduces the necessity of labelling spam messages and is employed to legitimate emails. The approach represents legitimate emails as word frequency vectors, thereby measuring deviations from the normal pattern.

Laorden et al. [133] proposed a spam filtering method for anomaly detection to reduce the necessity of labelling spam messages, employing the representation of one class of email (i.e., legitimate or spam). A data reduction algorithm is applied to the labelled dataset to reduce processing time, maintaining the detection rates and analysing the suitability of choosing a legitimate or spam email as a representation of abnormal behaviour. Goodman et al. [134] proposed a technique using a bipharite graph to give a score in anomaly detection by classifying a short message service as either spam or normal to detect lateral movement within a network. The above works were reviewed to identify ways to incorporate the technique into our intelligent filtering technique based on device-type. A behaviour-based network access control proposed in [65] uses network behaviour to provide access control policies. These security policies are updated over time to adopt network host behavioural changes as well as to determine the type of behaviour that can be accepted from network hosts in enterprise networks. The proposed work implements a behaviour-based profile using a voting process to detect and block behaviour that does not comply with enterprise policies.

2.3.4 Comparison of the closely related works

Table 2.3 provides a comparison of the closely related works in the field of fingerprinting and behaviour profiling techniques. These works use important attributes to fingerprint or profile devices, device types, device drivers, access points, service use or network traffic. Specifically, the fingerprinting techniques focus on the identification of network devices, such as device, device type, device driver, or access points, whereas the behaviour profiling techniques focus on the detection of abnormal patterns of network

devices. Both approaches are important, yet none of the related works considers combining them to develop a post-authorisation network access control technique that can be used to identify abnormal patterns based on device type. Hence, the proposed work combines the most important attributes used in fingerprinting and behaviour profiling techniques. These attributes include the device and device type used, the application used in generating the network traffic, e.g. the ping response from the devices, the location and the packet inter-arrival times. These attributes are considered here because they can help to identify the behaviour of the devices while accessing an enterprise network. For instance, the devices and device types are smartphones, tablets and laptops, which can be easily identified, while the applications can be ping and iPerf. The location can be identified from the access points, and the packets can be determined by the network traffic speed rate and payload size. In addition, the traffic type can be identified from active and passive network monitors, and the inter-arrival time is the time lapse between the two consecutively received frames.

Table 2.3 A comparison review of the closely related works

Approach		Detection Type	Features Used
Jana et al.	[93]	Device Driver Fingerprinting	Time delta
Radhakrishnan et al.	[97]	Device and Device type Fingerprinting	Inter-arrival time
Xu et al.	[98]	Device Fingerprinting	Transmitted signal frames and location
Dalai et al.	[99]	Device type Fingerprinting	Probe request frames
Kim and Kim	[71]	Service use pattern	device type, access time, access location and use time
Kang et al.	[40]	Service use pattern	Access time, location and time of use
Frias-Martinez et al.	[109]	Network profiling	flows, average flow size, average flow duration, packets in all flows, average number of packets per flow, IP address and average packet size
Proposed approach		Device-type profiling	Device type, application, traffic type, location and IAT

The aforementioned attributes and packet inter-arrival times are used in the proposed work because the latter creates a unique device signature that can be used to easily identify a device or device type. By monitoring inter-arrival times, it should be possible to distinguish individual devices or devices of the same type, i.e. devices that have the same hardware configuration. It has been discussed in the literature [135] that the packet inter-arrival time is a unique feature in which a device exhibits some traits

inherent to its physical implementation. As a result, it is assumed that packet creation varies across different device architectures and is influenced by the processor, direct memory access controller, memory and clock-skew. Similar variations exist between device architectures, which can be used to differentiate devices with the same hardware configuration. Therefore, device type can be easily configured and identified based on packet inter-arrival times, and abnormal patterns can be identified based on devices or device types, depending on the given problem or the issue that the researcher is addressing.

2.4 Chapter Summary

This chapter presented a literature review in the network access control domain. The chapter started by introducing NAC systems, their background, history and features. It also reviewed enterprise network security with the aim of determining and identifying the security requirements for BYOD enterprise networks. Then, a thorough investigation was conducted on the more prevalent security attacks on NAC to identify their limitations and the suitable solutions. The reviewed work focused on fingerprinting techniques, behaviour profiling and intelligent filtering techniques.

The fingerprinting techniques focus on reconnaissance (i.e. identification of devices or device-types) to collect information about a possible attack. Conducting reconnaissance is important, however, this technique does not provide the means to take any action after the identification of a possible attack. Also, while the research works presented in section 2.3.1 cover intensive research on fingerprinting techniques in mobile device networks to identify hosts, access points, network interface cards, devices and device-types, none actually address this problem. Our work in [3], however, did address this problem and proposed a device-type profiling technique. The devices and device-types are identified based on their packet inter-arrival time measurements.

Similarly, there are several publications in the literature in the field of behaviour profiling on mobile devices and network traffic. The research works presented in section 2.3.2 mostly focus on profiling user, location, calling activity, battery and network contextual data. While these are important, privacy is a major challenge in these approaches as they mainly focus on transparent and implicit authentication techniques. Also, privacy is another concern in network profiling approaches as sensible network contextual data are used for profiling. Another limitation is that none of these studies proposed a way to deal with the abnormal patterns identified. Our work in

[4] introduced a novel device-type profiling approach that detects abnormal patterns using packet inter-arrival times.

The last part of this chapter in section 2.3.3 investigates intelligent filtering techniques. These are techniques used in blocking abnormal patterns in network traffic. There has been intensive research into filtering approaches, however, most of these techniques focus on anomaly detection, using a filter that blocks abnormal patterns. This chapter has investigated these techniques to identify ways to incorporate them into our device-type profiling approach, which will use artificial intelligence techniques to develop a mechanism that can block abnormal patterns based on device-type profiling.

Chapter 3

Artificial Intelligence for Outlier Detection

Having underlined the importance of network access control systems in BYOD enterprise networks and highlighted the associated security issues, the thesis next introduces the artificial intelligence techniques and algorithms used to respond to these security challenges, for example, unauthorised access and data leakage, among others. Most of the response techniques identified here are data mining and machine learning models, which are categorised into supervised, unsupervised, and semi-supervised learning models. Outlier detection techniques and classification techniques are discussed in section 3.1, which also justifies choosing the clustering-based outlier detection and neural network algorithms. Next, the clustering-based outlier detection techniques are introduced in section 3.2, while section 3.3 concerns the neural network algorithm and the evaluation metrics used in this research. A summary concludes the chapter in section 3.4.

3.1 Outlier Detection and Classification Techniques

The significant amount of data that flows through mobile devices makes enterprise networks vulnerable. Attackers are using these vulnerabilities to break into organisations to cause data leakage and to gain illegal access to enterprise data. AI techniques are applied to address these challenges as physical devices such as sensors and actuators cannot provide sufficient protection in this regard [26]. Artificial intelligence provides numerous techniques that are flexible and that have learning capabilities that assists in detecting and mitigating attacks. These techniques are referred to as Data Mining and Machine Learning approaches, and are used interchangeably [136].

Data Mining (DM) is the process of transforming raw data into useful information. DM algorithms focus on extracting knowledge from a large amount of data to discover patterns that help to understand the complex relationships within a given dataset. Another example is applying learning algorithms to extract the behavioural patterns from network traffic. These techniques aid in the development of predictive models that enable an adaptive security response team to identify the behavioural patterns of the devices in the network, especially in cases were employees perform work-related tasks using their personally owned mobile devices [137].

Machine Learning (ML) is an application of artificial intelligence that enables automatic learning and then applies that learning without the need for human intervention after gaining some particular knowledge from a dataset. ML is a subset of AI technique that provides computing-based resources with the ability to learn without being explicitly programmed for that particular purpose. Learning algorithms are implemented in ML to formally compute the process of automatic pattern recognition and intelligent decision making based on training sample datasets. ML algorithms are categorised into immune-based, symbol based, connectionist based and behaviour based, none of which have advantage over the others. Also, ML problems are solved using supervised, unsupervised or semi-supervised learning approaches [137].

3.1.1 Supervised Outlier Detection Techniques

The current setting in enterprise networks is that they are vulnerable to attack due to security flaws in the system designs and implementation. These flaws might be procedural errors, or code or design errors, among others [138]. Attackers exploit these security flaws or vulnerabilities using a series of techniques and sequence of events to help them break into their target enterprise network. The sequence of events that occur due to security flaws are referred to as attack patterns. These patterns can be used to prevent further attacks by applying supervised detection techniques to the network traffic datasets. The supervised machine learning approach uses a predefined series of network traffic datasets with labelled inputs and a target (known outputs) to build a detection model. The inputs are used to train a machine-learning algorithm to predict the output of the pattern that is, or is not, part of the training dataset. Supervised learning algorithms are used to solve problems where prior knowledge of the data exist and the data itself is labelled [139]. They are categorised into classification and regression models. Classification problems are used when the output variables are classified into categories (e.g. normal or abnormal) whilst regression problems are

used in cases where the output is a real value. Examples of such algorithms are neural networks, support vector machines, and logistic regression, among others [140].

3.1.2 Semi-supervised Outlier Detection Techniques

In semi-supervised outlier detection techniques, the network traffic datasets are assumed to have labelled instances for only normal class(es) and no labels in the anomaly class(es), respectively. Due to the large network traffic data that pass through enterprise networks, this type of outlier detection technique has been widely used to solve a variety of problems in cases where the dataset is partially labelled and cannot specify all the exact normal and abnormal class(es). Unsupervised learning algorithms are of great practical value because of their capability to alleviate the cost of having to render fully labelled training datasets, especially in cases where it is impossible to label all data instances [139], [141]. The algorithms used in semi-supervised outlier detection techniques are similar to those used in supervised and unsupervised algorithms as the aim is to bridge the gap between them [142]. This is also known as the one-class classification problem, for which a series of well-known algorithms are used to solve this kind of problem, which include the one-class support vector machine, kernel density estimation, and auto-encoders, among others [143].

3.1.3 Unsupervised Outlier Detection Techniques

In unsupervised learning outlier detection techniques, network traffic datasets are assumed to have only inputs without known outputs. Unsupervised outlier detection is introduced to address the problems with supervised techniques in real network environments. For example, in the case of unauthorised access to enterprise network resources, the combination of clustering-based outlier detection and deep learning algorithms allows for cybersecurity systems to have a greater degree of accuracy and confidence in what could be considered as a potential abnormal pattern(s). Since the network traffic datasets in the real network environments consist of both normal and abnormal traffic, unsupervised outlier detection techniques therefore use unlabelled data as input to find abnormal traffic buried in the network network traffic datasets without prior knowledge of the data labels. Subsequently, unsupervised outlier detection relies on the following assumptions: *"normal data covers majority while anomaly data are minor in network traffic flow or audit logs; anomaly data points or normal data points are similar in their identity groups while statistically different between groups"* [137]. This learning paradigm is considered an imbalanced learning problem [137]. The normal

and abnormal data can be clustered and the current solutions to unsupervised outlier detection problems are found using clustering-based outlier detection techniques [141], [143], [144]. Therefore, this research focusses on clustering-based outlier detection techniques in order to gain better insight into the datasets and apply step-by-step knowledge discovery processes to achieve all the objectives.

3.2 Clustering-Based Outlier Detection

Clustering-based outlier detection is categorised into clustering-based and K-Nearest neighbour-based techniques. The rationale behind clustering-based outlier detection is to train datasets with a clustering algorithm to learn the normal and abnormal behavioural patterns of the data. Clustering-based outlier detection techniques can be classified into density-based, hierarchical-based and partitioned-based clustering [141]. Density-based clustering can produce outlying points along with a normal cluster. The hierarchical-based clustering identify the close clusters and continue merging them until they are all merged. Partitioning techniques divide the data instances into multiple partitions where each partition is referred to as a cluster. Partitioning-based methods have certain advantages because exciting patterns and structures can be found directly from large datasets with little background knowledge [145].

K-Nearest neighbour-based clustering algorithms can be categorised into two groups [137]. The first group consists of K-means clustering and self-organising map-based algorithms, whilst the second are density-based algorithms such as CLIQUE and MAFIA [145]. Density-based algorithms are computationally intensive and their anomaly detection results are not good, especially in cases where the data has varying densities [137], [141], [146]. Therefore, the following sections will focus on clustering-based algorithms due to their particular advantages.

3.2.1 K-means Clustering

K-means clustering is one of the most popular techniques used for outlier detection. It is fast, robust and relatively efficient [115]. It is also easy to understand, and can be used with iterative refinement to produce improved results when the dataset is distinct, or data are well separated from each other [137]. The K-means clustering technique is an iterative algorithm that partitions the dataset into k pre-defined distinct non-overlapping subgroups called clusters where each data point belongs to only one group [147]. The K-means clustering algorithm tries to group the intra-cluster data

points to as similar as possible while keeping the different clusters to as far as possible. Also, it assigns similar data points to a cluster such that the sum of the squared distance between the data points and the arithmetic mean of all the data points that belong to that cluster such that the less variation within clusters the more similar the data points are within the same cluster.

K-means clustering algorithm iterates between these two steps until no data points change the individual clusters. The iterations are performed after the algorithm initialises and estimates cluster k parameters, and randomly assign centroids for each cluster k. The algorithm iterates between two steps, data assignment and the centroid update step. For the data assignment, the data is assigned to its closest centroid. More formally, if cp is the collection of centroids in D (device) in set IAT, then each data point x is assigned to cluster k based on $cp \in$ D $dist(cp, x)^2$, where $dist =$ the Euclidean distance and the i^{th} cluster centroid is denoted by S_i. Then the centroid update step is recomputed using the mean of all the data points and assigned to the cluster centroids to optimise the function of K-means [148], denoted as:

$$IAT = \frac{1}{S_i} \sum X_i \in S_i{}^{X_i} \tag{3.1}$$

Moreover, to ensure that the centroid update step and data assignment guarantees there will not be any data point change in the clusters and that the maximum number of iterations is reached [2]. One of the major challenges in clustering is determining the optimal number of clusters, k [115]. The correct choice of k is usually not clear, with interpretations depending on the scale and shape of the data points and the desired clustering analysis. Alternatively, the optimal choice of k should strike a balance between the maximum compression of the data using two or more clusters. If an appropriate value of k is not apparent from prior knowledge of the properties of the data, it must be chosen randomly (i.e. $k = 2, 3, 4, 5...n$) based on the problem at hand. There are several methods can be used for determining the optimal number of k [149]. Example of these methods are: elbow method, Davies Bouldin index, Silhouette index, Dunn index, Partition Coefficient, among others. For example, Davies Bouldin (DB) index was used in [150],[151], [152],[153] to determine the optimal number for k, by calculating the intra-cluster similarities and inter-cluster differences to produce a set of clusters with an index for each cluster parameter [154]. They stated that the DB-index values helps them in determining the number of clusters k; based on the smallest DB-index values they have identified. Also, DB index is the best criterion for specifying the optimal number of k [155].

3.2.2 Cluster-Based Local Outlier Factor (CBLOF)

This outlier detection technique follows a global approach in which the outlier score is assigned to each instance of the entire dataset. He et al. [156] developed (CBLOF) a measure to identify outliers using a squeezer algorithm. The squeezer algorithm takes input from a dataset and sets a unique identifier for each tuple in the data to produce cluster results. The first tuple in the data is read in, a cluster structure is formed, and the other tuples are read iteratively. The similarity function is used to compute the similarities for each cluster and embodied within the corresponding cluster structure. The most significant similarity value is measured and compared against a threshold. If the threshold falls within the range of smaller or larger cluster centroids, then the values are added to the cluster with the highest similarity. The cluster structure continues updating each tuple in the data until all the tuples have been traversed.

CBLOF assigns an outlier score based on the distance to the nearest Large Cluster (LC) multiplied by the size of the cluster C_j, the object to which it belongs. From the equation, point p lies in the Small Cluster (SC) and the score would be equal to the distance to C_i which is the nearest LC multiplied by five which is the size of the cluster in SC. The squeezer algorithm partitions the data into large and small clusters and the Find CBLOF operation calculates the outlier score using two parameters: α, which is set to have a value of 0.9 to 1.0, whilst the authors recommend assigning a static value of 5 to β. α specifies the percentage of the non-outlying dataset and β specifies the boundary (b) between the large and the small clusters. The anomaly score is computed by the distance (t, c_j) of each instance to its cluster centre multiplied by the instances belonging to its cluster. For small clusters, the distance to the closest larger cluster is used. The procedure of using the number of cluster members as a scaling factor is used to estimate the local density of the clusters (t).

Goldsteine et al. [143] identified the fact that the use of cluster density as a scaling factor might result in incorrect density estimation. They accordingly developed unweighted-CBLOF (uCBLOF) using the K-means clustering algorithm and CBLOF outlier detection. The cluster density estimation was thus removed, and the outlier results were found to be better than for CBLOF. Duan et al. [157] proposed an improvement of CBLOF using local density instead of clustering. The proposed Cluster-Based Outlier Factor (CBOF) used Local Density Based Spatial Clustering of Applications with Noise (LDBSCAN) to detect outliers and assign clusters to Local Outlier Factors (LOF). However, their outlier detection algorithms are computationally extensive in large datasets and it takes considerable time to produce a result, and choosing the threshold can be difficult if the dataset is not well understood.

3.2.3 Local Density Cluster-Based Outlier Factor (LDCOF)

This uses the local outlier detection approach to detect outliers that are ignored by global approaches, especially in cases where there are varying densities within a dataset. The LDCOF addresses CBLOF's shortcoming by estimating the local density with an average distance for all cluster members to the nearest centroid and by assuming a spherical distribution of the cluster members. Amer et al. [144] proposed LDCOF using the same approach as CBLOF. The only difference is that an outlier score has been added to the data instances, where an outlier score of 1.0 or below is assigned to normal instances. Likewise, a score above 1.0 is assigned to abnormal instances.

LDCOF estimates the local density from the average distance of all cluster members from the centroid. In general, this seems a better estimation than CBLOF, but it is still not perfect. In this case, there are two circumstances that might lead to a bad estimation of the density: (1) in the case of having non-spherical distributions, the average radius of the cluster can be misestimated, especially for very long-shaped ellipsoids, normal instances at the long ends might get too large an anomaly score and outlying instances close to the long side might also be incorrectly estimated as normal; and (2) outliers far from the centroid point tend to increase the average distance drastically. When the average distance is large, the local outliers are not found any more since they are considered to be normal.

3.2.4 Clustering-based Multivariate Gaussian Outlier Score (CMGOS)

CMGOS is an enhancement of LDOF that uses a Gaussian model to get an improved estimation of local density for outlier detection. When the clustered data is connected to a CMGOS operator, the data is assumed to originate from a Gaussian distribution. Then, the algorithm calculates the local density using a fixed number of clusters to find the best fit for the assumed Gaussian distribution or determine the number of clusters based on a Bayesian approach. The outliers' scores are computed according to the centroid and the multivariate Gaussian of the cluster. Then, the local density estimation is performed by estimating a multivariate Gaussian model and the divergence, such as squared euclidean distance, mahalanobis distance, and squared loss, among others, can all serve as a basis for computing the anomaly score [158].

The anomaly score is computed by dividing the divergence of an instance to its nearest centre using a distribution fitted with a certain confidence interval as a normalisation process, where an outlier score of ≤ 1.0 indicates a high probability

of the instance being normal. The anomaly score can be estimated using reduction, which is similar to the multivariate Grubb's test, regularisation, which is similar to classification, and the minimum covariance determinant (MCD), which has idea of estimating a compact covariance matrix via a brute-force search for normal records. The CMGOS algorithms works according to the following steps:

Algorithm 3.1 CMGOS Algorithm [158]

```
1  procedure CMGOS
2      SET Data X to {x1 ... Xn}
3      CALL clustering Algorithm with Data X
4      Input cluster output C = {C1, C2, ..., Ck}
5          SET threshold P → 0.95-1.0
6          SET threshold y → 0.01-0.05
7      SELECT reduction, regularisation or Minimum Covariant
          Determinant (MCD) in CMGOS Operator Menu
8          IF MCD is selected THEN
9              SPECIFY sample according to the probability of
                  normal class
10         END IF
11     SPECIFY the number of time to remove outlier and to
          recompute the covariance matrix
12         Compute distance for all instances x to cluster
              centroid
13 end procedure.
```

3.3 Neural Network

Neural Networks (NN) is a type of supervised learning algorithm that attempts to simulate a network of neurons as inspired by the working function of a human brain. This approach has been used to solve complex classification and regression problems. Neural networks are very well known due to their ability to identify and detect complex non-linear relationships among input variables. Neural network algorithms consist of an input, hidden, and output layers. The neural network is defined as an algorithm; however, Howard Rheingold [159] defined NN as a kind of technology, not an algorithm, that has weights on it such that the weights can be adjusted so that it learns. The network is taken through trials until the desired model is achieved. The input, hidden, and output layers are illustrated in Figure 3.1, which shows how each neural network

layer is connected to the next to get the desired output. The input layer is responsible for feeding the input variables provided whilst training the network. The hidden layer consists of neurons used to process the input variables using activation functions to translate the input values to output values.

The most well-known functions in NNs are the unit step function, linear function, sigmoid function, and the hyperbolic tangent function (tanh), among others. These activation functions are mathematical equations that are used to determine the output of a neural network, and each function is attached to a neuron to determine the network activation based on the relevance of the model's predictions. Also, the activation functions are used to normalise the output of each neuron between the range 0 and 1 and -1 and $+1$. The elements in each layer of neural networks are highly connected by connections based on numeric weights that are learned by the algorithm. The output layer is responsible for predicting the class for a given input according to the weights defined through the hidden layer. There are many types of NNs such as feed-forward, back-propagation, multi-layer perceptron, and many more, all of which function in an analogous manner to the nervous system in the body [160]. Due to the large amount of data that passes through a network and larger neural networks to work with, deep learning neural networks were introduced to improve the performance of neural networks in general [161]. Due to the nature of this research, the kind of data, and the problems defined in the objectives, a deep learning neural network algorithm was used to solve the related problems. The specific algorithm used is part of a recurrent neural network called Long Short Term Memory (LSTM).

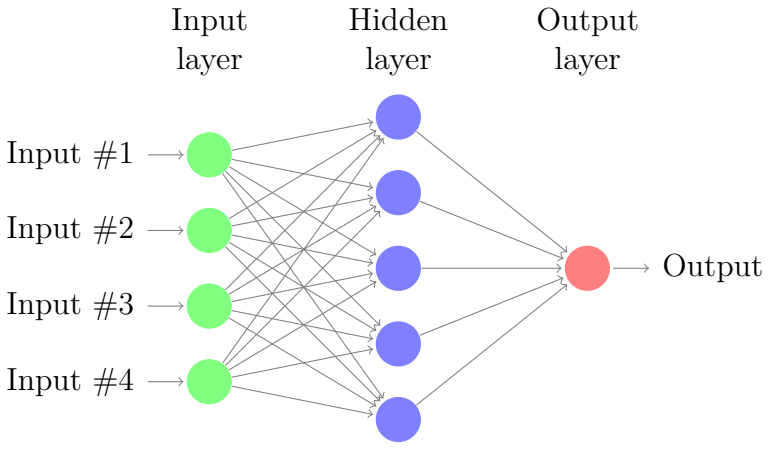


Fig. 3.1 Neural Network Diagram

3.3.1 Long Short Term memory Networks

LSTM was proposed by [162] as a solution to the vanishing gradient problem to preserve the errors that can be backpropagated through time and layers. LSTM maintains more constant errors which allows recurrent networks to continue to learn over multiple time steps in order to remotely open the channels that link causes and effects within a network. LSTMs are designed to remember information for a long period of time in order to avoid long-term dependency problems where they easily learn and remember the behaviour of a network. The LSTM form a chain of repeating modules of neural network modules that initialises the weight matrices and bias terms. The LSTM in Matlab toolbox consists of three options such as Nonlinear Auto-regressive with External Inputs (NARX), Nonlinear Auto-Regressive (NAR) and Nonlinear Input-output that can be used to solve nonlinear time series problems [162]. These architectures ends with a fully connected layer and a regression output layer. Also, the addition of softmax layer between the fully connected layer and the regression output can help to predict class labels.

The LSTM layers consists of input, hidden and output layers. The input layer consists of an input $x(t)$ device-type profile input and a delayed input $y(t)$. The hidden layers operate in gated cells so that information can be stored in, written to, and read out from the cell. The cell state consists of four gates, namely the gate, forget, input and output gates. A gate consists of a sigmoid activation function, which is similar to tanh activation, and which uses values between 0 and 1 to update or forget. The forget gate reads from the delayed input ($y(t-1)$ to make the decision to identify the abnormal pattern from $y(t-1)$ and keep the relevant pattern. It is associated with a sigmoid function that forwards the input and target values or variables (0, 1) to output gates to complete the flow of information throughout the gates, which can be represented as:

$$x_t = \sigma(W_x \cdot [y_{t-1}, f_t] + b_x) \tag{3.2}$$

The above equation multiplies the forget gate by the previous state such that when the values 0×0 are identified as abnormal inter-arrival time points. In the input gate, we pass the new values from the previous hidden state to the present cell state through a sigmoid function so that the sigmoid layer decides on the values that update the inputs and the tanh layer functions to create the vectors for new candidates being added to the new cell state. The input gate is represented as:

$$i_t = \sigma(W_i \cdot [y_t - 1, f_t] + b_i) \tag{3.3}$$

$$\tilde{C_t} = \tanh(W_C \cdot [y_t - 1, f_t] + b_C) \tag{3.4}$$

Where i_t is the previous input cell and $\tilde{C_t}$ is the updated input cell, and to get the present cell state multiplied by the previous cell state x_t, forgetting the abnormal patterns. Then, we add i_t and $\tilde{C_t}$. These are the new vectors, scaled by the delays to update each cell state value, as represented by:

$$C_t = y_t * C_t - 1 + i_t * \tilde{C_t} \tag{3.5}$$

The final step in the hidden layer is the output state that decides on what hidden state will do using the previous input to predict the next output by multiplying C_t by tanh to block values between $(-1, 1)$ and then multiply the result with the output of the sigmoid function so that the output of the filtered IAT traffic is displayed in the output layer. The output gate is represented as:

$$o_t = \sigma\left(W_o[x_t - 1, b_t] + b_o\right) \tag{3.6}$$

$$x_t = o_t * \tanh(C_t) \tag{3.7}$$

Then, the final output layer presents the output results of the model, including the number of values in the training, validation and testing. It displays the classification accuracy

3.3.2 Neural Network Performance Metrics

Machine learning classification problems are generally evaluated in two stages, namely training and testing. In the training stage, also called the learning process, the evaluation metrics of the classification algorithm are used to optimise the classification algorithm to select the optimal solution that produces the most accurate results, whereas in the testing stage, the evaluation metrics are used to measure the effectiveness of the classifier when tested with the unseen data. These classification metrics fall into different categories such as threshold type, Area Under Curve (AUC), Hybrid, and Mean Square Error (MSE), among others. The threshold type is used for solving two-class problems, where the classification model decides over a set of objects that can be expressed in a 2×2 matrix, where the rows indicate the actual class and the columns represents the predicted class [163]. The example of threshold type includes accuracy, recall, precision, and F-score, among others [164]. The AUC evaluation metrics are used to optimise learning models and also to compare learning algorithms

[165], whilst the hybrid is used for building an optimised heuristic classifier and the MSE measures the differences between the predicted and the desired outcome of a model and is used for evaluating data with continuous variables. As such, the best performance metric depends on the problem domain and the existing studies have not compared these metrics due to their complexity [166]. The most commonly used performance evaluation metrics for evaluating neural network algorithms are classified into metrics for evaluating classification and regression models.

Classification Evaluation Metrics: The most commonly used performance metrics in classification are based on confusion matrices, such as accuracy, defined as the number of correctly classified data points among all the data points. Precision and recall are respectively defined as the percentage of positive points within all positively labelled data points and the fraction of correctly classified data points of a particular class within all data points that belong to that class. The F-score is the weighted harmonic mean of precision and recall. Meanwhile, the Specificity (SPC), True Negative Rate (TNR) and False Positive Rates (FPR) are respectively defined as the number of correct negative predictions divided by the total number of all negatives, the number of incorrect negative predictions divided by the total number of all negatives, and the number of incorrect positive predictions divided by the total number of all negatives [164]. These evaluation metrics provide quantifiable evidence of how effective an algorithm is at classifying data patterns as normal or abnormal.

Regression Evaluation Metrics: Regression metrics are used to determine the linear relationships between the dependent Y_i and independent (predictor) variables, Z_1 [167]. These relationships are broadly used to predict the behaviour of the output response variables along with changes in the prediction variables and their MSE values. The lower the MSE values, the better the result. Also, R values are used to determine the relationship between the two variables in which an R value of 1 means a close relationship whilst 0 means a random relationship.

3.4 Chapter Summary

In summary, this chapter introduces all the algorithms used in this thesis. The chapter starts by introducing the artificial intelligence technique in cybersecurity that was used in outlier detection techniques. It also introduced outlier detection and classification techniques in the network access control domain. These techniques were discussed and

how they are applied to solve similar kinds of problems was described in detail in the literature review. Clustering-based techniques are the well-known techniques used to solve this kind of problems. Therefore, special attention has been given to clustering-based outlier detection techniques. These techniques were introduced along with the justification for selecting the technique used herein. The other section introduced the network classification algorithm that will be used to validate the performance of clustering-based data classification and long short term memory networks for building the intelligent filtering technique described in the research objectives.

Chapter 4

Data Analysis using K-means Clustering

In the previous chapter, we described and justified the existing outlier detection techniques suitable for this research; therefore, this chapter identifies and analyses a suitable dataset. The dataset identified was provided by the Georgia Institute of Technology. It contains the packet inter-arrival time values of 27 mobile devices captured from active, passive and isolated network monitors. K-means clustering is one of the most commonly used exploratory data analysis techniques to analyse and gain insight into data structures to group similar data points into subgroups known as clusters. Section 4.1 justifies our dataset selection and description, and then section 4.2 presents the experimental settings, while 4.3 analyses the clustering results for the devices in each dataset. After that, a device-type profiling is prepared in section 4.4 and section 4.5 contains a summary of the chapter.

4.1 Dataset Selection and Description

This research relies on an existing dataset consisting of network traffic traces from mobile devices such as smartphones, tablets and laptops. The objective of the research is to analyse the network traffic traces from these devices to define normal and abnormal profiles and subsequently develop a technique that can classify abnormal profiles. Therefore, this section outlines the dataset selection, description and analysis presented in this chapter.

4.1.1 Dataset Selection

In the selection of a suitable dataset for this research, several dataset repositories, such as the Community Resource for Archiving Wireless Data at Dartmouth, (CRAWDAD), Centre for Applied Internet Data Analysis (CAIDA), Outlier Detection Datasets (ODDS), DARPA, and Measurement and Analysis on the WIDE Internet (MAWI), among others, are researched to find suitable data to satisfy the requirements of this research. Our selection of a dataset is based on the important attributes (e.g. time, device-type, application used and location) suitable for defining and identifying abnormal from normal patterns. We have compared the datasets and chosen the one that satisfies our requirements. For example, the DARPA dataset [168] has been widely used by the research community to solve anomaly detection problems; however, in our case, the devices used to generate the data are not up to date and we are at the IoT edge. Therefore, using this dataset would not add sufficient value to this research. The ODDS dataset repository [169] also has a collection of datasets available for anomaly detection problems. While there are enough datasets in these repositories, none of these are suitable for the purposes of this research as the datasets were generated randomly from network devices, making it difficult to differentiate the data from smartphones, tablets and laptops; the same is true for the Centre for Applied Internet Data Analysis (CAIDA) [170] datasets. Another problem with the CAIDA dataset is that a lot of packets are lost, which would have a significant effect on this research. The closest dataset we have identified is the MIT reality dataset [171]; however, this dataset is also not up to date, whereby the devices used were 2004 models, and its main focus is on mobile device usage and movements.

Finally, the Gatech fingerprinting dataset (GTID) [97] was identified from the Crawdad repository [172]. This contains the packet inter-arrival time measurements of 27 mobile devices, including smartphones, tablets and laptops, connected through access point observation for different protocols (e.g. TCP, UDP and ICMP) and directions. Therefore, this dataset is suitable for use throughout this thesis. It was chosen because it contains the most important attributes needed for this research; examples of the attributes are smartphone, tablet, and laptop, location, time, and application. Moreover, it is the only one among the researched datasets to specify whether the data were captured from smartphones, tablets, laptops, or gaming devices, among others. While ODDS, MAWI, DARPA, etc. contain the target amount of information for device-type profiling, the essential requirement for use with BYOD is that the dataset needs to be specifically captured directly from smartphones, tablets and laptops to address the research problem.

4.1.2 Gatech Dataset Description

The characteristics of the datasets are device type, packet inter-arrival time, packet payload size, network traffic speed rate, application, and traffic type. The device types are smartphones, tablets and laptops. The packet inter-arrival time is the unique signature generated by each device type. The speed rates and payload sizes are represented in three cases (cases 1, 2, and 3), respectively. The packet payload size for cases 1 and 3 are 64-byte packets at a speed rate of 1 megabit per second (Mbps), while the packet size for case 2 is 1400 bytes at a speed rate of 8 Mbps. The applications used are iPerf, Ping, SCP, and Skype, and the protocols used are TCP, UDP and ICMP, respectively. In addition, the traffic type is represented based on the traffic monitors (active, isolated and passive network traffic monitors) used for capturing the inter-arrival time values for each device.

As stated earlier, the dataset contains the inter-arrival times of the device types measured via active, isolated and passive network traffic datasets. In capturing the network traffic, the authors in [97] state that they generated the data considering various attack scenarios. These attack scenarios were introducing constant delays to the packet stream, injecting random delays into the packets, tunnelling packets through another protocol, loading the CPU with intensive applications to overshadow normal behaviour, varying the packet size, and changing the data rate, among others. These attack scenarios were included when capturing the data and caused anomalies in the packet inter-arrival time capture while making it look like the devices had been attacked by an attacker with knowledge of the behaviour of the devices accessing the network. This is the main reason why the datasets contain anomalies. Moreover, after capturing the datasets, the authors did not specify where the attack originated as well as which device types or network traffic types contained the abnormal patterns. Also, the datasets were not labelled as having normal or abnormal inter-arrival time points based on the attack scenarios mentioned by the authors. This is one of the greatest limitations of the research and leads us to make use of the k-means clustering technique to gain insight into the data and to identify and separate the normal from abnormal inter-arrival time points; we also use CMGOS to label the normal and abnormal inter-arrival time points.

Moreover, the data contains multiple files (in the .mat format) suitable for use in MATLAB. We select and extract the data files on smartphones, tablets and laptops and convert them into Comma Separated Value (CSV) format. The reason for choosing smartphones, tablets and laptops is that our research objectives focus on NAC devices, and these are the only devices supported; therefore, other devices were not considered.

The files are converted into CSV because most data mining tools, such as RapidMiner Studio [173], do not support the .mat file extension. The files are then organised according to $filename \rightarrow Application \rightarrow Protocol \rightarrow Case$ (e.g. iPerf-TCP-Case 2). An overview of these datasets is presented in Tables 4.1, 4.2 and 4.3.

Table 4.1 An overview of the devices in the Active Network Traffic Datasets

Application	Protocols	Case No.	Payload Size	Speed Rate	Device Types
Ping	ICMP	1	64 bytes	1mbps	10 Acer Netbooks, 10 Asus Netbooks, 8 Gateway Netbooks , 2 Google Phones, 2 Lenovo Laptops and 2 Asus Tablets.
Ping	ICMP	2	1400 bytes	8mbps	10 Acer Netbooks, 10 Asus Netbooks, 8 Gateway Netbooks , 2 Google Phones, 2 Lenovo Laptops and 2 Asus Tablets.

Table 4.2 An overview of the devices in the Isolated Network Traffic Datasets

Application	Protocols	Case No.	Payload Size	Speed Rate	Device Types
Iperf	TCP	2	1400 bytes	8mbps	5 Dell Netbooks, 3 iPad's, 2 iPhone 3G, 2 iPhone 4G and 2 Nokia Phones.
Iperf	UDP	1	64 bytes	1mbps	5 Dell Netbooks, 3 iPad's, 2 iPhone 3G, 2 iPhone 4G and 2 Nokia Phones.
Iperf	UDP	2	1400 bytes	8mbps	5 Dell Netbooks, 3 iPad's, 2 iPhone 3G, 2 iPhone 4G and 2 Nokia Phones.
Iperf	UDP	3	64 bytes	1mbps	5 Dell Netbooks, 3 iPad's, 2 iPhone 3G, 2 iPhone 4G and 2 Nokia Phones.
Ping	ICMP	1	64 bytes	1mbps	3 Dell Netbooks, 3 iPads, 2 iPhone 3G, 2 iPhone 4G and 2 Nokia Phones.
Ping	ICMP	2	1400 bytes	8mbps	3 Dell Netbooks, 3 iPads, 2 iPhone 3G, 2 iPhone 4G and 2 Nokia Phones.
Scp	TCP	1	64 bytes	1mbps	5 Dell Netbooks, 3 iPad's, 2 iPhone 3G, 2 iPhone 4G and 2 Nokia Phones.

Table 4.3 An overview of the devices in the Passive Network Traffic Datasets

Application	Protocols	Case No.	Payload Size	Speed Rate	Device Types
Iperf	TCP	1	64 bytes	1mbps	10 Acer Netbooks, 10 Asus Netbooks, 8 Gateway Netbooks , 2 Google Phones, 2 Lenovo Laptops and 2 Asus Tablets.
Iperf	UDP	1	64 bytes	1mbps	10 Acer Netbooks, 10 Asus Netbooks, 8 Gateway Netbooks , 2 Google Phones, 2 Lenovo Laptops and 2 Asus Tablets.
Iperf	UDP	3	64 bytes	1mbps	10 Acer Netbooks, 10 Asus Netbooks, 8 Gateway Netbooks , 2 Google Phones, 2 Lenovo Laptops and 2 Asus Tablets.
Ping	ICMP	1	64 bytes	1mbps	10 Acer Netbooks, 10 Asus Netbooks, 8 Gateway Netbooks , 2 Google Phones, 2 Lenovo Laptops and 2 Asus Tablets.
Scp	TCP	1	64 bytes	1mbps	10 Acer Netbooks, 10 Asus Netbooks, 8 Gateway Netbooks , 2 Google Phones, 2 Lenovo Laptops and 2 Asus Tablets.
Iperf	UDP	2	1400 bytes	8mbps	2 Lenovo Laptops and 1 Asus Tablets.
Skype	UDP	1	64 bytes	1mbps	2 Lenovo Laptops and 2 Asus Tablets.

The active dataset presented in Table 4.1 contains the packet inter-arrival time points of 68 devices, namely ten Acer Netbooks, ten Asus Netbooks, eight Gateway Netbooks, two Google Phones, two Lenovo Laptops and two Asus Tablets in two different cases. The isolated dataset presented in Table 4.2 contains the packet inter-arrival time points of 94 mobile devices, namely five Dell Netbooks, three iPads, two iPhone 4G, two iPhone 3G and two Nokia Phones in three different cases and applications. The passive dataset presented in Table 4.3 contains the inter-arrival time points of 245 devices, including ten Acer Netbooks, ten Asus Netbooks, eight Gateway Netbooks, two Google Phones, two Lenovo Laptops and two Asus Tablets in the different cases and applications, respectively. The applications used for generating

these datasets are iPerf and Ping and are divided into three cases (cases 1, 2 and 3) mentioned above according to different packet size/rate settings.

4.2 Packet Inter-Arrival Time Data Analysis

The packet inter-arrival time is the time between each arrival of a packet into the system and the arrival of the next packet. Hence, it is commonly used to measure the incoming and outgoing packets in a network, which is one of the fundamental characteristics of internet traffic. Packet inter-arrival time measurements are integral parts of network traffic management and monitoring and control tasks in packet-switched networks [174]. Packet inter-arrival time is important in this research because it can help to understand the behaviour of devices or device-types based on their packet inter-arrivals. Moreover, we analyse these packet inter-arrival time values for each device using the K-means clustering algorithm described in section 3.2.1 and following the experiment settings presented in the next sections.

4.2.1 Data Analysis Experiment Settings

We pre-process the datasets in RapidMiner Studio [173], a data science and machine learning platform for data science processes, such as data preparation, machine learning, deep learning, text mining, and predictive analytics [175]. The operators needed to perform K-means clustering in RapidMiner Studio are: (1) load data operator, (2) K-means clustering operator, and (3) cluster model visualiser. Each operator is connected to the next, has its own configuration settings (see the connections in Figure 4.1, and is broken down with the background processes in Figure 4.2). The load data operator can access the datasets stored in the repository and load them into the process. The K-means operator has certain parameter configuration settings that can be used to partition the dataset values according to the number of clusters. These parameters are K, Maximum runs, Measure type, Divergence and Maximum optimisation steps. K is used to determine the number of clusters, e.g. $K = 2, 3 \ldots n$. Maximum runs and maximum optimisation steps are by default 10 and 100, respectively, but can be adjusted based on user preference. Measure type consists of Mixed, Nominal, Numerical and Bregman divergence, and it works hand-in-hand with the divergence option so that when the measure type changes, the options in the divergence change. For example, when we select nominal values, the divergence can be set to the nominal distance; when we select Bregman divergence, the divergence has many options that can be used

to calculate the cluster distance. The well-known examples of the cluster distance measures are squared Euclidean distance, Mahalanobis distance, and squared loss, among others, and these are available under the divergence tab. The last operator helps in visualising the clustering results and captures the essential details of each cluster along with the Davies–Bouldin performance index for each cluster.

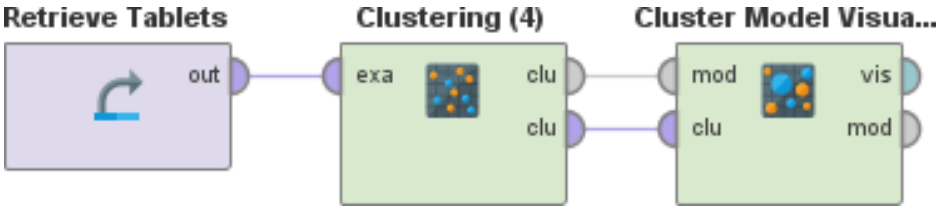


Fig. 4.1 K-means operator configuration Rapidminer

4.2.2 Determining the Number of Clusters

To determine the number of clusters, we experiment with different cluster measures, such as the silhouette index, Dunn index, elbow method, and Davies–Bouldin index, and examples of these results are presented in Tables A.1, A.2 and A.3 in Appendix A. The perfect settings that fit our datasets and give the best results are K = 2, maximum runs = 10, Measure type = Bregman divergence, Divergence = squared Euclidean distance and Maximum optimisation steps = 100. Also, the Davies–Bouldin index is used to determine the number of clusters due to its advantages over the other measures, including its simpler computation and the fact that the index is computed to measure the ideal number of clusters for each dataset, among others. Examples of these results are shown in Tables 4.4. The results in all the tables clearly show that the optimal number is $k = 2$ as the Davies–Bouldin index increases with increasing k. Therefore, the optimal number of k for solving this research problem is $k = 2$, which is used throughout the experiments.

Therefore, in the remaining experiments, we define and configure the K-means clustering algorithm based on the above justification. Our K-means clustering algorithm uses a squared Euclidean distance function to compute the distances (i.e., similarities) between the two clusters for each device and device-type to produce two clusters, i.e., $C0$ and $C1$. We assume the device and device-type to be represented by a set of vectors $Dev_D, Dev_{DT} = IAT_1, IAT_2, ..., IAT_n$, where D is the device, DT is the device-type and IAT is the data point distribution for each device or device-type. The clustering algorithm distributes the inter-arrival time values into k according to distance d and produces two clusters, cp. The squared Euclidean distances are calculated between

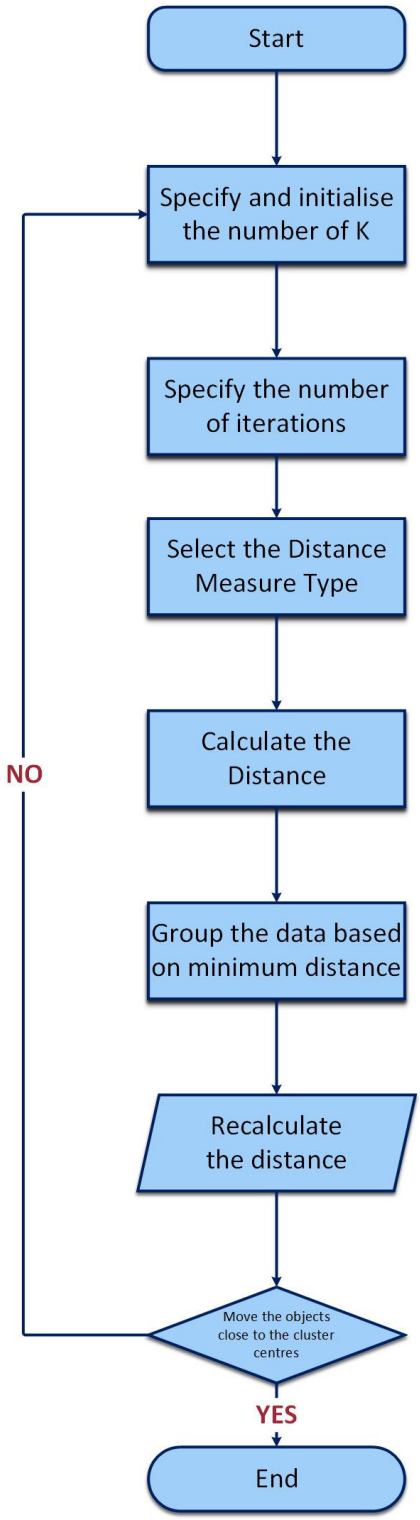


Fig. 4.2 K-means operator configuration

the clusters in a weighted Euclidean mode using an inverse of the average proportions as weights. Suppose c_j denotes the j^{th} element of the average of each cluster; in such a case, the squared Euclidean distance, denoted by E, between the two clusters $v = [v_1, v_2....v_j]$ and $nv = [nv_1, nv_2....nv_j]$ can be defined as:

$$E_{v,nv} = \sqrt{\sum_{j=1}^{j} \frac{1}{c_j}(v - nv)^2} \tag{4.1}$$

Where v is the varied cluster and nv is the non-varied cluster for each device and device-type. The squared Euclidean distance function computes the square root of the sum of the squares of the differences between the inter-arrival time values for each device or device-type. Note that all these features contribute equally to the function value. The inter-arrival time distributions are used to discover the impact of inter-arrival time variation from similar devices or device-types on the same network traffic.

Table 4.4 Davies-Bouldin index for Active, Isolated and Passive network traffic datasets

Network Traffic Type	Device	$k=2$	$k=3$	$k=4$	$k=5$
Active	Acer	0.020	0.060	0.075	0.235
	Asus	0.010	0.102	0.296	0.287
	GatewayNB	0.100	0.216	0.369	0.420
	GoogleP	0.509	0.541	0.642	0.649
	Lenovo	0.140	0.420	0.433	0.513
	Asus Tablets	0.409	0.438	0.492	0.565
Isolated	Dell	0.291	0.330	0.349	0.386
	iPads	0.307	0.367	0.393	0.416
	iPhone 3G	0.082	0.312	0.398	0.415
	iPhone 4G	0.021	0.358	0.387	0.393
	Nokia	0.169	0.228	0.236	0.248
Passive	Acer	0.043	0.328	0.360	0.381
	Asus	0.010	0.336	0.372	0.377
	GatewayNB	0.047	0.312	0.329	0.365
	GoogleP	0.002	0.364	0.414	0.437
	Lenovo	0.058	0.413	0.441	0.477
	Asus Tablets	0.553	0.548	0.627	0.674

4.2.3 Analysis of Clustering Results

The clustering results are analysed based on the inter-arrival time points associated with the mean of each cluster, and the descriptive analysis of the clusters and are presented in sections 4.3 and 4.4. The clustering analysis helps in the creation of clusters that can be used to identify and analyse the relationships between the inter-arrival time points for each device. For example, cluster centre analysis can be used to group the inter-arrival time points to partition the data and identify abnormal inter-arrival time points. It may be possible to use cluster centre analysis to solve many unsupervised machine learning problems.

Descriptive statistics is a term given to data analysis techniques that describe the main features of the data in a meaningful way so that patterns can emerge from that data. To examine the potential positive features of the clusters, we apply descriptive statistics to each cluster using quartiles to identify any significant data patterns. A quartile is a statistical term used in descriptive analysis when describing a division of observations. A quartile measures the spread of values below and above the mean by dividing the distributions into four defined intervals (minimum, first quartile, median, third quartile and maximum) based on the values of the data and how they compare with the entire set of observations. A quartile divides data into three points, namely the lower quartile (Q1), the median quartile (Q2), and the upper quartile (Q3) to describe the data. The lower quartile is the middle number between the smallest number and the median of the dataset. The median is the middle value of the data, and the upper quartile is the middle value between the median and the highest value in the data.

4.3 Cluster Centre Analysis

The inter-arrival time points associated with the mean of each cluster for the active, isolated and passive network traffic datasets described in section 4.1.2 is analysed. The resulting means of the clusters for each device and device-type are presented in Tables 4.5, 4.6, and 4.7. The tables contain the number of clusters, represented as $C0$ for the first cluster and $C1$ for the second cluster, which are the total number of inter-arrival points associated with the mean of the first and second clusters, respectively. Moreover, the descriptive analysis of the data in the cluster distributions of the individual devices and their device-types are also presented in the tables, which are presented and analysed based on notched box plots in section 4.4.1.

4.3.1 Active Network Traffic Dataset

The active network traffic datasets consist of the inter-arrival time values of the mobile devices described in the active network traffic dataset analysis. These mobile devices are Acer Netbooks (AC1-10), Asus Netbooks (AS1-10), and Gateway Netbook (GW1-8), Lenovo Laptops (L1-2), Google Phones(G1-2) and Asus Tablets (T1-2), and their clustering results are presented below. The clustering results for Acer Netbooks presented in Table 4.5 show that the devices AC1-10 have several inter-arrival time points in the normal and abnormal clusters ($C0$ and $C1$), and $C0$ has more inter-arrival time points than $C1$ for all the devices and device-types. For example, AC1 has $393,116$ inter-arrival time points in $C0$ and 61 inter-arrival points in $C1$. The $393,116$ inter-arrival time points for $C0$ are associated with the cluster centre 0.009s, and the 61 inter-arrival points for $C1$ are associated with the 0.935s cluster centre, which is similar for other devices and their device-types. These cluster centres are the mean values of each data partition, divided into two points for the devices and their device-types, respectively ,which can help to obtain insight into the data and identify the outlying inter-arrival time points. In the same table, the mean value of the normal cluster $C0$ for all the devices AC1-10 and their device-type Acer Netbook is 0.009s. Also, in $C1$, the cluster centre for AC1-10 and Acer Netbook, the minimum mean value among all the devices is 0.701s for AC6, while the maximum is 0.984s for AC7.

The devices AS1-10, GW1-8, G1-2, L1-2 and T1-2 and their device-types are analysed below and their results are presented in Tables A.4 and A.5 in Appendix A. The mean value of $C0$ for both devices and their device-types presented in the tables is 0.009s, while their associated inter-arrival time points are similar to AC1-10 above. The cluster centre for the abnormal cluster $C1$ for AS1-10 is observed with a minimum mean value of 0.721s and a maximum of 0.957s. Additionally, the minimum and maximum inter-arrival time values of $C1$ for GW1-8, G1-2, L1-2 and T1-2 fall between 0.087 and 1.008s. There are cases where some devices, such as AS7, have a mean value of 5.053s and influence the device-type with the same value. After a thorough investigation, by removing this value and repeating the experiment, the value is identified as an outlier and removing it improves the maximum value of both the device and the device-type, leading to a maximum value of 0.936s; the same applies to AC5, AS8, AS7 and L1, although L1 does not influence the device-type. Since this chapter is more concerned with the clusters, the data are left as they are so that the clustering-based multivariate outlier score algorithm in the next chapter can automatically detect it.

The above results inform us that the inter-arrival times of the normal cluster are similar and that there are outliers in the abnormal cluster that need further

Table 4.5 Descriptive analysis of device-types of Ping-ICMP-Case1 in active network traffic datasets

Device	Cluster	IAT Points	cluster centre(s)/Mean	Q1	Median	Q3	Min	Max	Std
AC1	C0	393,116	0.009	0.008	0.009	0.010	0.000	0.437	0.005
	C1	61	0.935	1.001	1.006	1.014	0.509	1.046	0.157
AC2	C0	394,563	0.009	0.008	0.009	0.010	0.000	0.425	0.005
	C1	31	0.973	1.002	1.006	1.015	0.510	1.649	0.189
AC3	C0	396,468	0.009	0.009	0.009	0.009	0.000	0.289	0.002
	C1	63	0.706	0.391	0.908	1.008	0.381	1.050	0.308
AC4	C0	394861	0.009	0.008	0.009	0.010	0.000	0.426	0.005
	C1	32	0.937	1.001	1.006	1.011	0.504	1.037	0.161
AC5	C0	394,897	0.009	0.008	0.009	0.010	0.000	0.268	0.003
	C1	63	0.716	0.392	0.874	1.007	0.382	1.038	0.302
AC6	C0	396,172	0.009	0.008	0.009	0.009	0.000	0.322	0.003
	C1	53	0.701	0.394	0.661	1.006	0.383	1.019	0.292
AC7	C0	397,615	0.009	0.009	0.009	0.009	0.000	0.412	0.004
	C1	29	0.984	1.005	1.007	1.012	0.633	1.026	0.084
AC8	C0	397,360	0.009	0.009	0.009	0.009	0.000	0.356	0.002
	C1	62	0.710	0.393	0.759	1.005	0.381	1.019	0.289
AC9	C0	397,457	0.009	0.009	0.009	0.009	0.000	0.472	0.004
	C1	54	0.953	1.003	1.007	1.011	0.512	1.051	0.148
A10	C0	397,095	0.009	0.008	0.009	0.010	0.000	0.416	0.004
	C1	31	0.949	1.002	1.007	1.011	0.496	1.016	0.143
ACER	**C0**	**3,959,761**	**0.009**	**0.008**	**0.009**	**0.009**	**0.000**	**0.472**	**0.004**
	C1	**326**	**0.957**	**1.003**	**1.006**	**1.010**	**0.496**	**1.649**	**0.140**

investigation. The results of this investigation are used in the next step of this research. To gain additional insight into the above results and to validate a device-type profiling approach, we further analyse the clustering results using notched box plots in section 4.4 below.

4.3.2 Isolated Network Traffic Dataset

The isolated network traffic datasets consist of seven data files, each file containing inter-arrival time traffic data for Dell Netbooks (DN1-5), iPads (IP1-3), iPhone 3G (IT1-2), iPhone 4G (IF1-2) and Nokia Phones (NP1-2). Table 4.6 show example of the clustering results for these mobile devices and their device-types. The clustering results show the mean value for each cluster. The mean values for the normal cluster

$C0$ is 0.001 s. The clustering results for the Dell Netbooks presented in Table 4.6 show that $C0$ has more inter-arrival time point than $C1$. For example, the mean values of the normal cluster for DN1 is 0.001. The total number of inter-arrival time points is $841,299$ whereby $C0$ has $840,955$ inter-arrival time points and $C1$ has 344. The other devices, DN2-5, have more inter-arrival points in $C0$ and $17-344$ inter-arrival time points in $C1$, while their mean values fall between 0.007 and 0.428s, respectively. Also, the mean values for $C0$ for IP1-3, IF1-2, IT1-2 and NP1-2 and their device-types are 0.001s, and the mean value of $C1$ ranges from 0.007 to 2.660s.

Table 4.6 Descriptive analysis of Dell-Netbooks of iPerf-TCP-Case 2 in isolated network traffic datasets

Device	**Cluster**	**IAT Points**	**cluster centre(s)/Mean**	**Q1**	**Median**	**Q3**	**Min**	**Max**	**Std**
DN1	C0	840955	0.001	0.001	0.001	0.001	0.000	0.058	0.001
	C1	344	0.132	0.087	0.134	0.175	0.082	0.187	0.044
DN2	C0	1327118	0.001	0.001	0.001	0.001	0.000	0.037	0.001
	C1	320	0.074	0.052	0.061	0.082	0.038	0.284	0.039
DN3	C0	1288629	0.001	0.001	0.001	0.001	0.000	0.053	0.001
	C1	66	0.111	0.083	0.088	0.172	0.062	0.177	0.042
DN4	C0	2557115	0.001	0.000	0.001	0.001	0.000	0.004	0.000
	C1	26530	0.007	0.004	0.005	0.008	0.004	0.202	0.004
DN5	C0	3059230	0.001	0.000	0.000	0.001	0.000	0.176	0.001
	C1	17	0.428	0.268	0.447	0.543	0.233	0.657	0.154

Moreover, the results for the other devices and their device-types (IP1-3, IF1-2, IT1-2 and NP2) in the isolated network traffic datasets are presented in Tables A.8 - A.12 in Appendix A. In the tables, the mean values of the normal cluster $C0$ are 0.001s and for the abnormal cluster $C1$ they lie between 0.016 and 0.122s. In some cases, the values fall between 4.189 and 59.876s. These results show that the inter-arrival times for the normal cluster are all similar, yet there are changes in the abnormal cluster that need further investigation.

4.3.3 Passive Network Traffic Dataset

The passive network traffic dataset consists of eight data files. Each file contains inter-arrival time network traffic data for the similar, Asus Netbooks, Gateway Netbooks, Google Phones, Lenovo Laptops and Asus Tablets. The mean values for the normal cluster are 0.001, 0.002, 0.009 and 0.011s. An example of the clustering result of Gateway Netbooks is presented in Table 4.7. As can be seen from the table, the mean

value of the normal cluster $C0$ for all devices and their device-types is 0.011s. The mean values for $C1$ for all the devices and their device-types have minimum and maximum values of 0.0128 and 0.135s, respectively. This result shows that the devices and their device-types have similar mean values in the normal clusters and different values in the abnormal clusters. Additionally, the mean values for $C0$ for the other devices and devices types in the same dataset are presented in Table A.13 and remain at 0.011s. Also, the minimum and maximum mean values of $C1$ for G1-2, L1-2 and T1-2 fall between 0.016 and 0.021s and 0.0128 and 0.135s for AC1-10 and AS1-10, respectively.

Table 4.7 Descriptive analysis of device-types of iPerf-UDP Case 3 in passive network traffic datasets

Device	Cluster	IAT Points	Cluster Centre(s)	Q1	Median	Q3	Min	Max	Std
GW1	C0	320880	0.011	0.011	0.011	0.011	0.000	0.069	0.002
	C1	355	0.135	0.132	0.135	0.138	0.076	0.143	0.005
GW2	C0	320913	0.011	0.011	0.011	0.011	0.000	0.070	0.003
	C1	334	0.131	0.131	0.134	0.138	0.073	0.152	0.015
GW3	C0	320929	0.011	0.011	0.011	0.011	0.000	0.065	0.002
	C1	312	0.134	0.132	0.135	0.138	0.074	0.146	0.007
GW4	C0	320803	0.011	0.011	0.011	0.011	0.000	0.069	0.004
	C1	367	0.128	0.131	0.134	0.138	0.070	0.165	0.020
GW5	C0	320886	0.011	0.011	0.011	0.011	0.000	0.071	0.002
	C1	311	0.135	0.132	0.135	0.138	0.080	0.152	0.006
GW6	C0	320925	0.011	0.011	0.011	0.011	0.000	0.069	0.002
	C1	353	0.135	0.132	0.135	0.138	0.077	0.155	0.005
GW7	C0	320848	0.011	0.011	0.011	0.011	0.000	0.072	0.002
	C1	355	0.135	0.132	0.135	0.138	0.076	0.148	0.006
GW8	C0	320874	0.011	0.011	0.011	0.011	0.000	0.072	0.002
	C1	309	0.135	0.132	0.135	0.138	0.077	0.158	0.005
GatewayNB	**C0**	**2567066**	**0.011**	**0.011**	**0.011**	**0.011**	**0.000**	**0.072**	**0.002**
	C1	**2688**	**0.134**	**0.132**	**0.135**	**0.138**	**0.072**	**0.165**	**0.010**

The results for the other datasets are presented in Tables A.14 - A.16. As mentioned earlier, the mean values of the normal cluster, $C0$, for all datasets are 0.001, 0.002, 0.009 or 0.011s, depending on the dataset. The normal cluster purely shows the mean values for each dataset, and the results clearly demonstrate that the measurements for each dataset are different. The minimum and maximum mean values for the abnormal cluster, $C1$, for most of the devices and device-types in the remaining datasets fall between 0.016 and 1.633s, respectively. In a few cases, we observe some devices having

large values and influencing their device-type, such as the Acer in Table A.21, which shows a mean value of 27.747s and 42.855s for Asus in Table A.19. This result does not mean that these are abnormal inter-arrival time points but rather shows that they have the greatest mean values. Therefore, these datasets will be investigated further to identify abnormal profiles and provide the means with which these can be identified. Moreover, the results for the device-types are not significantly different from the results for their individual devices.

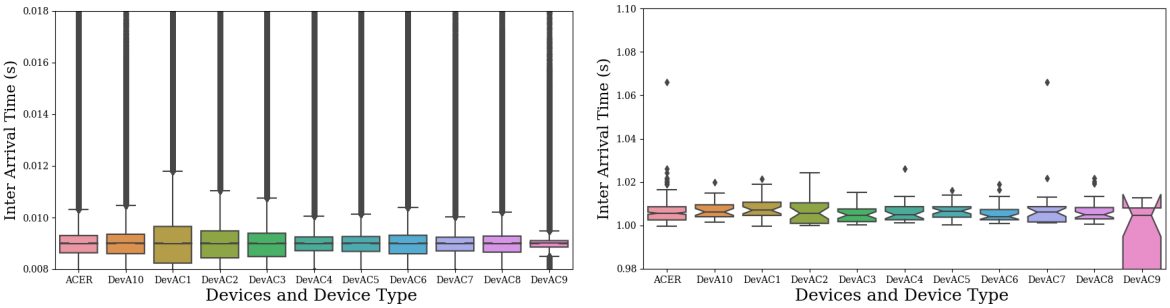
4.4 Notched Box Plot Analysis

The notched box plots of the clusters for the active, isolated and passive network traffic datasets are analysed in this section. Notched box plots are used to gain insight into each cluster to identify how their inter-arrival time values are distributed and to validate whether device or device-type profiling is a valid approach [176]. This can be determined when the notched boxes overlap. The notched box plots are a very convenient visualisation of the statistical five-tuple, which consists of the quartiles and the minimum and maximum data values. For each dataset, a box is drawn from the first quartile to the third quartile, and the median is marked with a thick line. Additional whiskers extend from the edges of the box towards the minimum and maximum of the dataset, but no further than 1.5 times the inter-quartile range. Data points outside the range of box and whiskers are considered outliers and are drawn separately as small circles. A variation of standard box plots is to add notches. Notches surround the median and roughly indicate the significance of differences between the values. If the notches of two boxes do not overlap, their medians are significantly different at a 95% confidence level.

Moreover, the notched box plots were also used to answer the third research question, namely whether device-type profiling is a valid approach. It may only be considered valid if the associated statistical assumptions are met, which would only be the case if the notches of the devices and their device types overlap. Overlapping notches would indicate that the inter-arrival time values in the devices are not significantly different from their device types; having met this condition, device-type profiling is considered as a valid approach within a 95% confidence level.

4.4.1 Active Network Traffic Dataset

In the notched box plot results for the 68 mobile devices in the active network traffic datasets, it was observed that all the notched boxes for the normal and abnormal clusters for devices and their device-types overlap. This gives us 95% confidence that the inter-arrival time values from the devices and their device-types are distributed within the same range. For example, the notched box plot of the normal cluster for the Acer Netbooks, illustrated in Figure 4.3a, shows that the notched boxes of the devices and their device-type overlap. These overlapping notches give 95% confidence that devices AC1-10 are from the same inter-arrival time distributions as their device-type (Acer Netbook). Similarly, the notched box plots of the abnormal cluster, illustrated in Figure 4.3b, overlap, indicating that AC1-10 and Acer Netbook are also from the same inter-arrival time distributions.



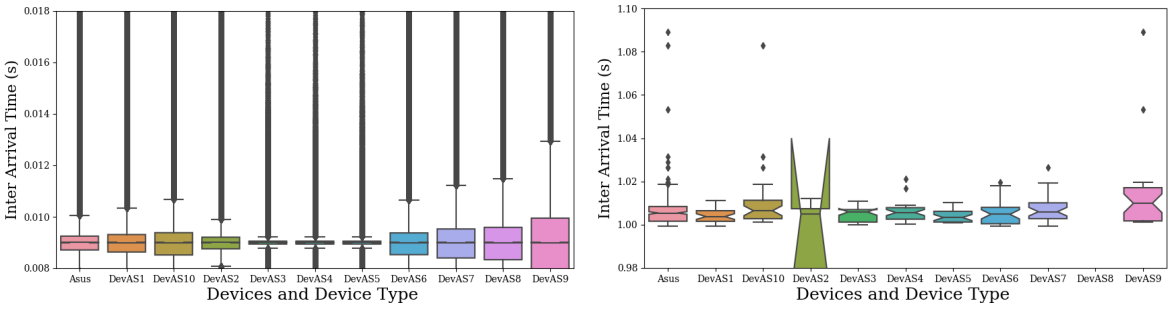
(a) Normal (b) Abnormal

Fig. 4.3 Notched box plots of normal and abnormal cluster for devices (AC1-10) and their device-type (Acer) in Active network traffic datasets

Another example of the normal cluster for the Asus Netbooks, presented in Figure 4.4a shows that the notched boxes of devices AS1-10 overlap with that of their device-type (Asus). The notches overlap in the abnormal cluster for the same devices and their device-types, as shown in Figure 4.4b. Thus, the results below show that the inter-arrival time distributions of the devices are similar to those of their device-types. Moreover, these results, and indeed the further examples presented in Figures A.1 - A.4 in Appendix A, gave 95% confidence that device-type profiling is a valid approach and can thus be implemented using active network traffic datasets.

4.4.2 Isolated Network Traffic Dataset

In the notched box plot results for the 94 mobile devices in the isolated network traffic dataset, it was observed that the notched boxes for the devices in the normal cluster



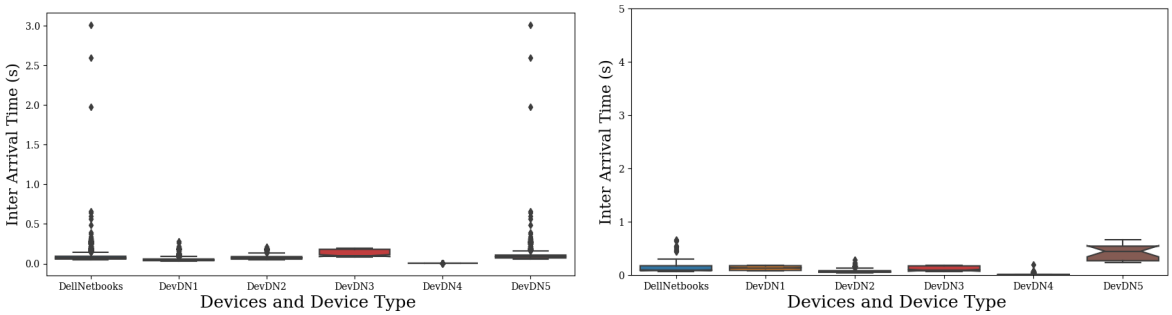
(a) Normal (b) Abnormal

Fig. 4.4 The notched box plots of normal and abnormal cluster for devices (AS1-10) and their device-type (Asus) in Active network traffic datasets

and their device-types overlap. The notched boxes shows that the inter-arrival time values of the devices and those of their device-type overlaps and lies within the 95% confidence level. For example, the notched box plots for the cluster of Dell Netbooks (DN1-5) presented in Figure 4.5a show that the inter-arrival time distributions of DN1-5 are similar to that of their device-type (Dell Netbook). Based on the figure, their inter-arrival time distribution values overlap, and as with the above, this gives us 95% confidence that the devices and their device-types are from the same inter-arrival time distributions. Another example where one of the devices in the abnormal cluster did not overlap with the rest of the devices and device-type is presented in Figure 4.5b. The figure shows that DN1-4 overlap with their device-type, but DN5 does not overlap with either the other individual devices or the device-type. Since the notched boxes of the normal cluster for DN1-5 and the abnormal cluster for DN1-4 overlap with their device-type, there is no confidence that all the devices and their device-types are from the same inter-arrival distributions; however, in this instance, DN5 represents an outlier that needs further investigation.

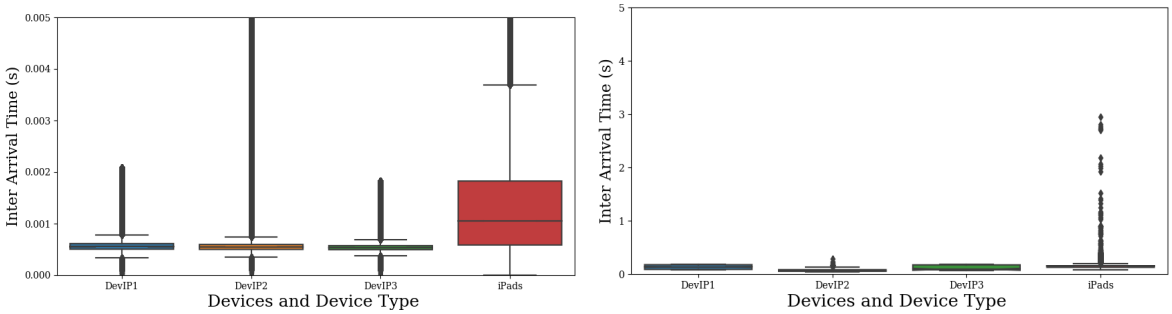
Figures 4.6a and 4.6b present other examples in which all the notched boxes of both clusters of iPads (IP1-3) overlap, with the associated 95% confidence that all the devices and the device-type are from the same inter-arrival time distributions.

Also, another example of Nokia Phones (NP1-2) in Figures 4.7a and 4.7b shows that the notched boxes of the normal cluster overlap, whereas the abnormal cluster for NP1 does not overlap with NP2 and the device-type. This result, based on the normal cluster, indicates that the devices and their device-type are from the same inter-arrival time distributions. This highlights that NP1 also needs further investigation. The results for all the devices and their device-types in the isolated network traffic datasets



(a) Normal (b) Abnormal

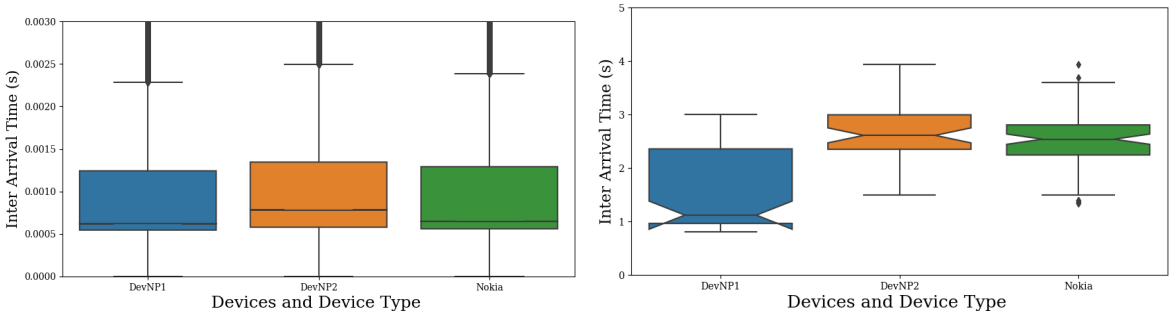
Fig. 4.5 The notched box plots of normal and abnormal cluster for devices (DN1-5) and their device-type (Dell Netbooks) in Isolated network traffic datasets



(a) Normal (b) Abnormal

Fig. 4.6 The notched box plots of normal and abnormal cluster for devices (IP1-3) and their device-type (iPads) in Isolated network traffic datasets

are not in any way distinct from the above results; further examples can be seen in Figures A.5 - A.6 in Appendix A.

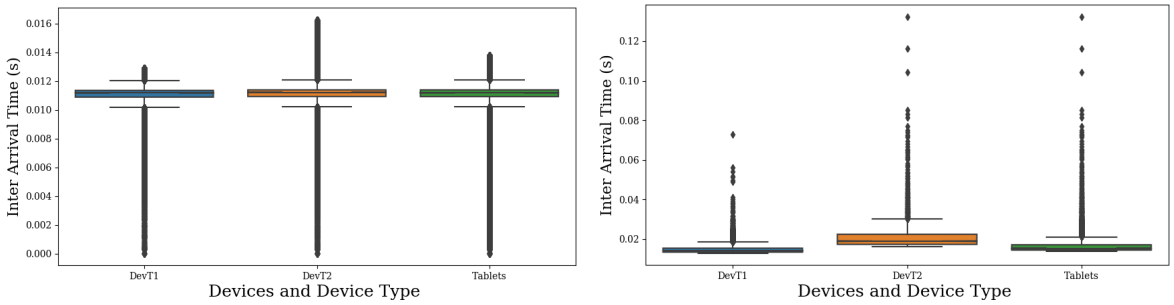


(a) Normal (b) Abnormal

Fig. 4.7 The notched box plots of normal and abnormal for (NP1-2) and their device type (Nokia Phones) in Isolated network traffic datasets

4.4.3 Passive Network Traffic Dataset

From the notched box plot results for the 245 mobile devices in the passive network traffic datasets, it was observed that the notched boxes of the normal cluster for the devices and their device-types overlap, giving 95% confidence that the data from the various devices and their associated device-types have similar inter-arrival time values. An example of the results for the passive network traffic datasets for a tablet, laptop and smartphone are illustrated in the figures below. According to Figures 4.8a and 4.8b, the inter-arrival time distributions for both clusters of the devices and their device-types lie within the same inter-arrival time ranges, implying a similar 95% confidence that the inter-arrival time values of the devices and their device-types overlap. Also, the results demonstrate that device-type profiling is a valid approach for this device-type.

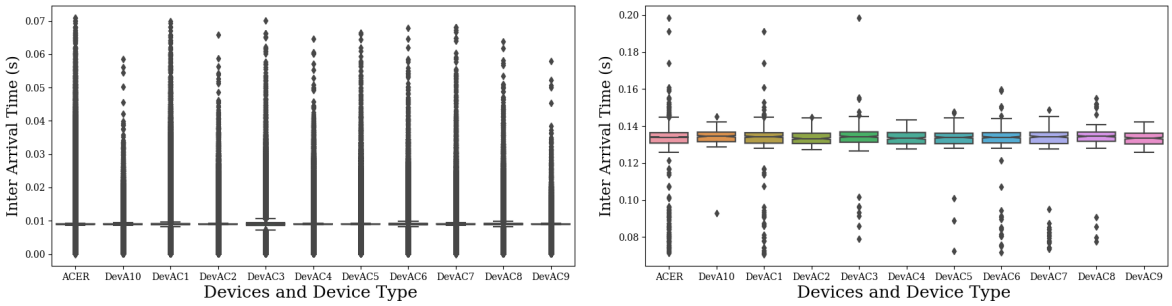


(a) Normal (b) Abnormal

Fig. 4.8 The notched box plots of normal and abnormal cluster for devices (T1-2) and their device-type (Asus Tablets) in Passive network traffic datasets

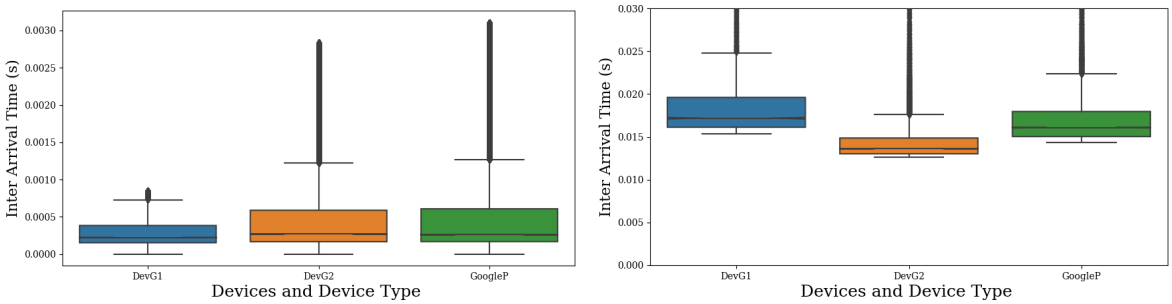
Moreover, similar results are observed for both clusters for the Acer Netbooks, as presented in Figures 4.9a and 4.9b. The outliers at the top and bottom of the devices and device-types do not affect the results because the notches overlap and suggest nothing that would otherwise invalidate our assumption. As outlier detection is one of the main objectives of this research, these outliers will further be investigated in chapter 5. Another example illustrated in Figures 4.10a and 4.10b, the notched boxes for the Google Phones (G1-2) overlap with that of their device-type. These results demonstrate that the devices and their device-type are from the same inter-arrival time distribution, with no visible outliers. Therefore, device type profiling is also a valid approach for this device-type.

In another example where one device has a large inter-arrival time value that influences the device-type. For example, in Figures 4.11a and 4.11b, AS9 has a large inter-arrival time value for one of its data points. After a thorough investigation, we note that the devices and their device-type are within the same inter-arrival time



(a) Normal (b) Abnormal

Fig. 4.9 The notched box plots of normal and abnormal cluster for devices (AC1-10) and their device-type (Acer) in Passive network traffic dataset



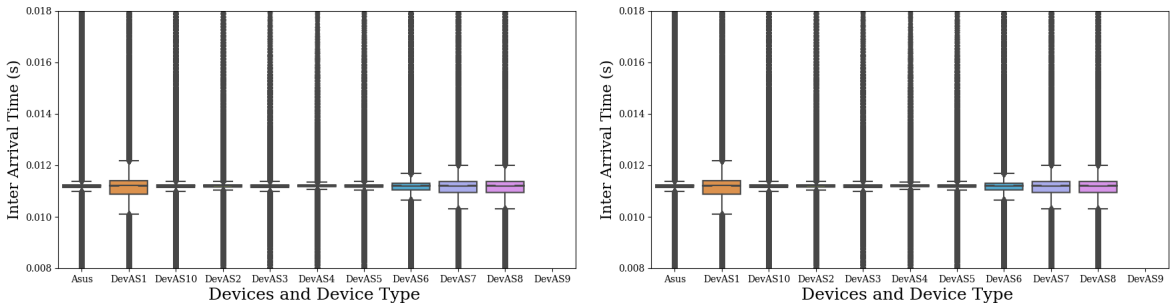
(a) Normal (b) Abnormal

Fig. 4.10 The notched box plots of normal and abnormal cluster for devices (G1-2) and their device-type (Google Phone) in Passive network traffic dataset

ranges. Therefore, these results demonstrate that device-type profiling remains a valid approach for the passive network traffic datasets as per the 95% confidence limits in that the devices and their device-types have similar inter-arrival values. Lastly, the results for the other datasets, presented in Figures A.7 to A.10 in Appendix A, suggest similar results, therefore, device-type profiling is a valid approach for all the device-types in passive network traffic datasets.

4.5 Chapter Summary

K-means clustering is an established technique for data analysis. This chapter presents a process of data analysis by applying K-means clustering to an existing dataset. Instead of analysing the data using the popular K-means approach (scatter plot), it used a notched box plot to provide additional insight into the data values. It started by analysing individual device data and later concatenated all the data from the same device-type into one data file and compared the results. The datasets were



(a) Normal (b) Abnormal

Fig. 4.11 The notched box plots of normal and abnormal cluster for devices (AS1-10) and their device type (Asus) in Passive network traffic datasets

concatenated to understand the relationships between the individual devices and their device-types as well as to demonstrate that their data distributions were similar. The results of the analysis presented in sections 4.3.1, 4.3.2, and 4.3.3 demonstrate that device-type profiling is a valid approach. Also, the results give 95% confidence that most of the distributions from the devices and device-types overlap. The clustering analysis helped to identify the normal and abnormal inter-arrival time points, which were then used to classify the data into normal and abnormal profiles. It should be noted that similar results were observed for the datasets presented in Tables A.4 - A.14; however, considering that each of the datasets would require a lengthy data analysis without providing additional information, the same data analysis can be applied to the other datasets. As a next step, we consider a device-type profiling approach in which the output of the clustering algorithm (clusters) is used as input for the clustering-based multivariate Gaussian outlier score (CMGOS) to classify the device-types into normal and abnormal profiles, respectively. The CMGOS settings and configurations used to classify the device-types into normal and abnormal profiles are presented in section 5.1.

Chapter 5

Device-Type Profiling Using Clustering-Based Outlier Detection

The device-type profiling discussed in this chapter is developed using the pre-processed data described and analysed in chapter 4. First, device-type profiling is defined in section 5.1, as are the device-type profiling algorithm and the experimental settings. Then, sections 5.2 and 5.3 present the device-type profiling and data labelling, respectively, both of which are achieved here using the clustering-based multivariate Gaussian outlier score (CMGOS) algorithm. The device-type profiling shows the percentages of the normal and abnormal inter-arrival time points. The data labelling labels the device-type datasets with either normal or abnormal inter-arrival time points depending on the outlier score so that the data can be useful in the identification of abnormal inter-arrival time points. Finally, a summary is presented in section 5.4.

5.1 Device-Type Profiling

Profiling is a technique for classifying and identifying the inter-arrival time pattern(s) that deviate from the rest of the inter-arrival time points. The profiling technique was applied to individual device-type to obtain a normal and abnormal device-type profile which can be use in identification of abnormal inter-arrival time point. In the device-type profiling implementation, the output of the k-means clustering algorithm was used as an input for CMGOS. The cluster centres help in the calculation of the multivariate Gaussian outlier score of each cluster based on the probability of how likely an inter-arrival time point is to be close to the cluster centre. For example, the inter-arrival time points associated with or close to the first cluster are added to the first cluster, with the same being true for all other clusters. Then, the CMGOS

calculates the outlier score for each inter-arrival time point based on the multivariate Gaussian outlier score of each cluster, and subsequently adds the outlier scores to each inter-arrival time point. An outlier score > 1.0 indicates a high probability of the instance being abnormal, whilst < 1.0 indicates a high probability of a normal inter-arrival time following the standard defined by Goldstein et al. [143].

5.1.1 Device-Type Profiling Algorithm

The device-type profiling algorithm was developed using the CMGOS operator available in RapidMiner Studio [173]. It combines the fast execution of k-means with the support of the Gaussian mixture model (GMM) to estimate the density of each cluster. The algorithm uses Euclidean distance as a distance measure to calculate the cluster centroids and the GMM in compacting the cluster into the group by partitioning the inter-arrival time points and adding them to the clustering groups they may belong to. Since we know the number of clusters beforehand (see section 4.2.1), we can segment the device-type data into two parts based on the number of clusters ($k = 2$), as justified in section 4.2.2, to determine the clusters for the normal and abnormal device-type profiles, whereby each data point local to each cluster is added to the closest cluster centre. For example, if a point is very close to the first cluster, it is added to the first cluster; otherwise, it is added to the second cluster. Then, we estimate the covariance matrix, $\sum_x$, of each cluster C_x based on the Euclidean distance, E_d. The covariance matrix for all inter-arrival times of C_x and are computed by reduction using the formula:

$$\sum_x : E_d(C_x) = \sqrt{C_x^n \times \sum_x -1 * C_x} \tag{5.1}$$

The $\sum_x$ estimation by reduction is an effective approach built into RapidMiner Studio to determine whether an instance is normal or abnormal using the probability P_n and the Chi-square distribution. This removes anomalies and recomputes $\sum_x$; it can be repeated several times, but a single iteration is usually sufficient [143]. Therefore, the output of the algorithm will add an outlier score to every inter-arrival time point. We then classify the inter-arrival time points based on the outlier score to obtain the normal and abnormal device-type profiles, add labels to the data, and visualise the results.

Algorithm 5.1 Device-Type Profiling algorithm

```
procedure CMGOS
    SET Data X to CONCAT (D1 ... Dn)
    CALL K-means clustering with Data X and number of cluster
       K SET to 2
    Input cluster output C = {C0,...,C1}
        SET threshold P → 0.99
        SET threshold y → 0.1
    SELECT reduction in CMGOS Operator Menu
        SPECIFY the number of time to remove outlier
        Recompute the covariance matrix
    Compute the Euclidean distance for all instances x to
       cluster centroid
    Add labels to all inter-arrival time points for each device
       -type
        IF outlier score <1.0 THEN
           SET label to normal inter-arrival time point
        ELSE
           SET abnormal inter-arrival time point
end procedure.
```

5.1.2 Device-Type Profile Experiment Settings

To develop a device-type profiling based on active, isolated and passive network traffic datasets, the experimental steps indicated in algorithm 5.1 are followed. This starts by first concatenating the datasets for each device-type and input them into the retrieve data operator, configure the k-means clustering operator using the same settings discussed in section 4.2.1. Then, configured the CMGOS operator based on the configurations that give the best results, namely the probability of normal = 0.99, gamma (γ) = 0.1, covariance estimation = reduction, time to remove outlier = 1 (i.e., the number of times the minimum covariance matrix removes outliers) and the measure type = mixed measure using mixed Euclidean distance. The local density estimation for each cluster is achieved using a multivariate Gaussian model. The Euclidean distance serves as the basis for computing the outlier score for each inter-arrival time point in the cluster. The covariance matrix for each cluster is then computed using two iterations to remove outliers and to recompute the covariance matrix.

The outlier score is then calculated by dividing the Euclidean distance of the inter-arrival time points nearest to each cluster centre to normalise the data with a certain confidence interval. The chi-square distribution is used in normalisation such that an outlier score <= 1.0 indicates a high probability of the inter-arrival time point falling within the normal profile. All these parameter settings are selected to build a device-type profile and label the inter-arrival time points for each device-type. The reason for choosing the squared Euclidean distance as a measure for both k-means and CMGOS is that it gives better results in terms of fast execution and accuracy compared to other distance measures [177]. Also, we label the inter-arrival time points based on their outlier scores by configuring the *generate attributes* operator with an expression (if (outlier $\geq$ 1, 'Abnormal', 'Normal')) to count and assign the outlier scores $>$ 1.0 into abnormal or, otherwise, normal profiles, respectively.

5.2 Analysis of Device-Type Profiling

The device-type profiling results for the sample datasets from the active, isolated and passive network traffic datasets is presented below. These samples can help in identifying the inter-arrival time differences for the device-types based on the active, isolated and passive network traffic measurements, hardware specifications, and model. The device-type profiling was applied to these different network traffic datasets to indicate the applicability of our approach.

5.2.1 Active Network Traffic

In the device-type profiling results for the device-types in the two active network traffic datasets, it was observed that the Acer, Asus and Gateway Netbooks have more inter-arrival time points than the Google Phone, Lenovo Laptop and Asus Tablet. During the experiments, it was noted that the laptops have more inter-arrival time points than smartphones and tablets, and they also have lower run times. This shows that there were more responses from the laptops, but this does not impact the effectiveness of our device-type profiling technique as the aim is to profile and label the inter-arrival time points of each device-type as either normal or abnormal profiles.

The device-type profiling results for the device-types in the first active network traffic dataset are presented in Figure 5.1. The figure shows that the normal device-type profiles for the Acer, Asus and Gateway Netbooks have 99.9% of the normal inter-arrival time points, and the Google Phone, Lenovo Laptop and Asus Tablet have

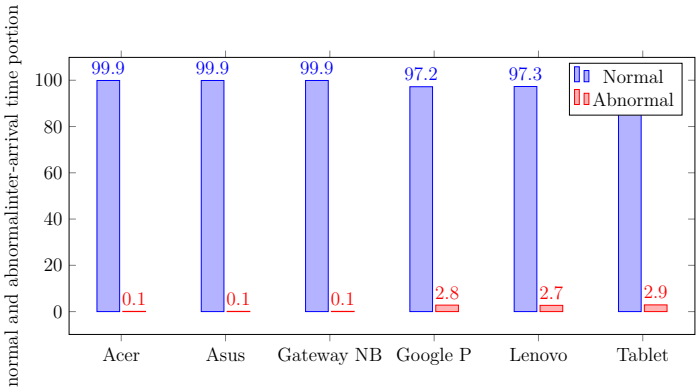


Fig. 5.1 The normal and abnormal device-type profiles of Ping-ICMP-Case 1 datasets

97% of the normal inter-arrival time points. The abnormal device-type profiles for the Acer, Asus and Gateway Netbooks have $0.1-0.3\%$ of the inter-arrival time points. In contrast, the abnormal device-type profiles for Google Phone, Lenovo Laptop and Asus Tablet have $2.7-2.9\%$ of the inter-arrival time points.

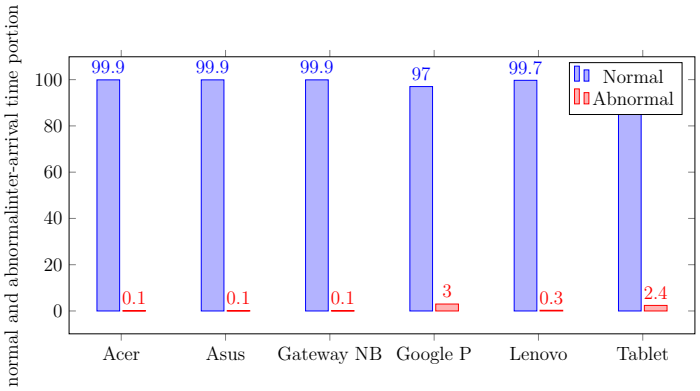


Fig. 5.2 The normal and abnormal device-type profiles of Ping-ICMP-Case 2 datasets

The device-type profiling results for the second active network traffic dataset presented in Figure 5.2 above show that the normal device-type profiles for the Acer, Asus and Gateway Netbooks and Lenovo Laptop have 99.9% of the inter-arrival time points, whereas the Google Phone and Asus Tablet have 97% of the inter-arrival time points. The abnormal device-type profiles for the Acer, Asus and Gateway Netbooks and Lenovo Laptop have $0.1-0.3\%$ of the inter-arrival time points, while the Asus Tablet and Google Phone have 2.4% and 3%, respectively.

The above results show that the majority of the device-types have few abnormal inter-arrival time points and also that the normal and abnormal device-type profiles

can be used to identify abnormal inter-arrival time points in active network traffic. The numeric values for these results are presented in Tables A.22 and A.23 in Appendix A.

5.2.2 Isolated Network Traffic

In the device-type profiling results for the device-types in the isolated network traffic datasets, it was observed that there is some fluctuations in the number of inter-arrival time points. For example, the laptops have more inter-arrival time points, but in some cases, the iPhones or iPad has more inter-arrival time points compared to the other device-types. However, this is of little importance if the objective of this chapter can be achieved, which is to profile and label the normal and abnormal patterns for each device-type.

The device-type profiling results for the device-types in the isolated network traffic datasets are presented in the figures below. In Figure 5.3, the normal device-type profiles for the Dell Netbook, iPad, and iPhone 3G have 99% of the inter-arrival time points. Also, the iPhone 4G and Nokia Phone have 97% of the inter-arrival time points. The abnormal device-type profiles for the Dell, iPad and iPhone 3G have 0.1% of the inter-arrival time points, while the iPhone 4G and Nokia Phone have 2.7% and 2.8% of the inter-arrival time points, respectively.

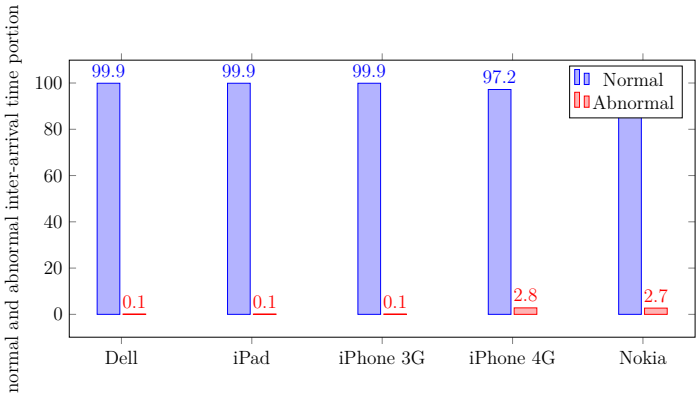


Fig. 5.3 The normal and abnormal device-type profiles of iPerf-TCP Case 2 datasets

In Figure 5.4 below, the device-type profiles for the Dell Netbook, iPad and iPhone 4G are shown to have 99% of the inter-arrival time points, while the iPhone 3G and Nokia Phone have 98%. The abnormal device-type profiles for the Dell Netbook, iPad and iPhone 4G have between 0.1 and 0.7% of the inter-arrival time points. Also, the abnormal device-type profiles for the iPhone 3G and Nokia Phone have 1.3% and 1.7% of the inter-arrival time points, respectively.

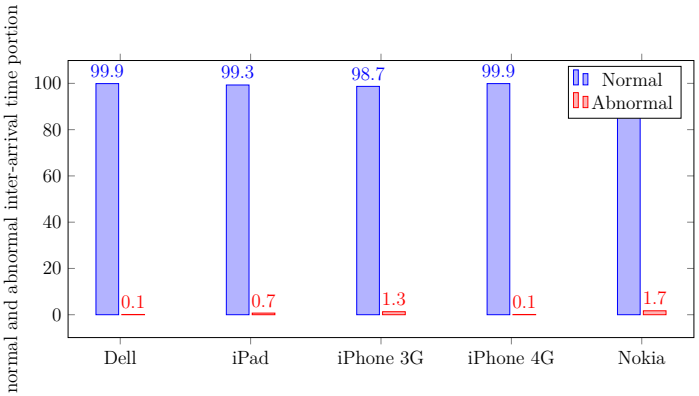


Fig. 5.4 The normal and abnormal device-type profiles of iPerf-UDP-case 1 datasets

The above results show that most of the device-types have less abnormal inter-arrival time points. Similarly, these results show no differences from the other device-type profiles of the other five isolated network traffic datasets presented in Figures A.11 - A.13. The numeric values for these results are also illustrated in Tables A.24 and A.30 in Appendix A.

5.2.3 Passive Network Traffic

The device-type profiling results for the device-types in the passive network traffic datasets presented below, we observe that some device-types have no abnormal inter-arrival time points. Also, in some cases, we note that the abnormal inter-arrival time points are greater than in the active and isolated network traffic datasets. In this section, we present the results for two datasets, each containing the device-type profiles for the Acer, Asus and Gateway Netbooks, Google Phone, Lenovo Laptop, and Asus Tablet in the passive network traffic datasets.

The device-type profile results in Figure 5.5 show that the normal device-type profile for the Acer and Asus Netbooks have 100% normal inter-arrival time points. These results show that not all networks have abnormal patterns, and there will be cases where the network has small, large or no abnormal patterns. For example, in the same datasets the Gateway Netbook, Lenovo Laptop and Asus Tablet have small abnormal patterns $(98.5 - 99\%)$ normal inter-arrival time points. Their abnormal device-type profiles contain between 0.4 and 1.5% inter-arrival time points.

The normal device-type profiles for the Acer, Asus and Gateway Netbooks, Google Phone, Lenovo Laptop and Asus Tablet presented in Figure 5.6 it was observed that five of the device-types have 99% of the inter-arrival points and the sixth device-type

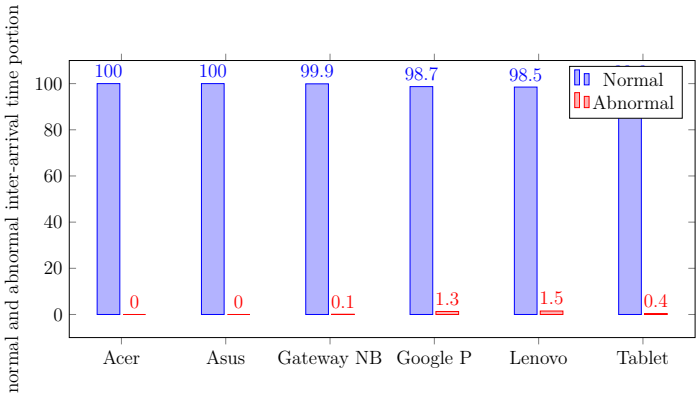


Fig. 5.5 The normal and abnormal device-type profiles of iPerf TCP Case 1 datasets

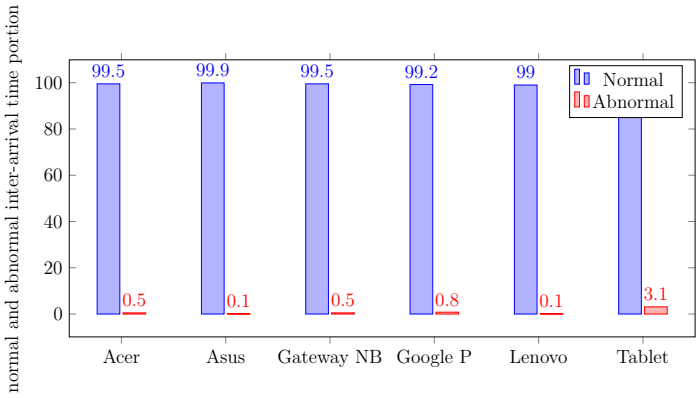


Fig. 5.6 The normal and abnormal device-type profiles of Ping-ICMP-Case 1 datasets

(i.e. Asus Tablet) has 96.9%. Whilst, the abnormal device-type profiles for Acer, Asus and Gateway Netbooks, Google Phone and Lenovo Laptop are 0.1% and for the Asus Tablet is 3.1%. The above results show that the majority of the device-types have few abnormal inter-arrival time points. Similarly, they show no significant differences to the other device-type profiles in the other six passive network traffic datasets presented in Figures A.14 - A.17. The numeric values of these results are also reported in Tables A.31 and A.38 in Appendix A.

5.3 Data Labelling

Data labelling is defined as the process of labelling data to allow machine learning algorithms to learn the patterns from that data. Data labelling is essential to machine learning algorithm predictions and classification problems. In labelling our data, we ensure that the inter-arrival time points for each device-type are labelled as being

normal or abnormal. These labels are obtained based on the outlier scores using the values identified in the device-type profiles. Our data labelling is achieved following step 8 of algorithm 5.1. In labelling the datasets, the cluster centres described in section 4.3 are used as a basis for separating the normal and abnormal profiles. For example, the mean value of the first cluster, $C0$, is 0.009s, and the mean value of the second cluster, $C1$, is 0.935s. Therefore, any value that falls into $C0$ is added to $C0$, and similarly for $C1$. The data labelling results for the active, isolated and passive network traffic datasets are presented in the following sections.

5.3.1 Data Labelling for Active network traffic dataset

The sample data labelling for the device-types in the active network traffic datasets are presented in Table 5.1. As can be seen from the table, the Acer Netbook has $3,968,592$ inter-arrival time points, comprising $3,965,611$ and $2,981$ normal and abnormal points, respectively. Recalling the centroid point results presented in Table 4.5, the same device-type is identified with one inter-arrival time point with a value of 5.744s in $C1$. This result shows that the CMGOS identifies more outliers that fall on or are close to $C1$. A similar situation exists for both the normal and abnormal inter-arrival time points for the Asus and Gateway Netbooks, Google Phone, Lenovo Laptop and Asus Tablet. Moreover, similar results were observed for the device-types in the active network traffic datasets presented in Tables A.22 and A.23. Since the algorithm can identify the inter-arrival time points for each cluster, we apply step 8 of algorithm 5.1 for all the inter-arrival time points to create the data labels for both normal and abnormal inter-arrival time points. For instance, the $3,965,611$ inter-arrival time points of the device-type 'Acer' are labelled as normal inter-arrival time points, while $2,981$ are labelled as abnormal inter-arrival time points. The algorithm takes $10-34$ seconds to identify the inter-arrival time points and label them into normal and abnormal profiles accordingly.

Table 5.1 Data labelling for Ping-ICMP-Case 1 Active network traffic dataset

Device Type	**IAT points**	**Normal points**	**Abnormal points**	**% abnormal**	**Run Time (s)**
Acer	3,968,592	3,965,611	2,981	0.01	34
Asus	3,969,874	3,967,493	2,381	0.01	32
Gateway NB	3,179,980	3,178,136	1,844	0.01	28
Google Phone	796,817	774,416	22,401	0.3	17
Lenovo	798,309	777,037	21,272	0.3	10
Tablet	794,975	772,017	22,958	0.3	23

5.3.2 Data Labelling for the Isolated Network Traffic Datasets

Data labelling was also applied to the device-types available in the isolated network traffic datasets. The example of the data labelling results presented in Table 5.2 shows that the results are similar to those for the active network traffic datasets. One of the differences observed is that there are more inter-arrival time points than in the active network traffic datasets. For example, the Dell Netbook has a total of $9,100,324$ inter-arrival time points, which is by far the most among the device-types presented in the active network traffic datasets. There are more abnormal inter-arrival time points identified than in the k-means clustering analysis. Also, similar results are observed for the device-types in the isolated network traffic datasets presented in Tables A.24 - A.30. The run time of the algorithm is between 19 and 1.35 seconds.

Table 5.2 Data labelling for iPerf-TCP-Case 2 Isolated network traffic dataset

Device Type	**IAT points**	**Normal points**	**Abnormal points**	**% abnormal**	**Run Time (s)**
Dell Netbooks	9,100,324	9,034,201	66,123	0.7	1.35
iPads	4,581,539	4,570,163	11,376	0.2	54
iPhone 3G	1,129,399	1,113,549	15,850	0.1	9
iPhone 4G	8,300,764	8,260,063	40,701	0.5	1.10
Nokia	1,563,011	1,558,888	4,123	0.03	19

5.3.3 Data Labelling for the Passive Network Traffic Datasets

The data labelling was applied to all the device-types in the passive networks traffic datasets. In labelling the datasets, we observe that some datasets have no abnormal inter-arrival time points, such as the device-types Acer, Asus, and Gateway Netbooks in Table 5.3. In contrast, the clustering analysis presented in section 4.3.3 shows less than 2000 abnormal inter-arrival time points across these device-types. After comparing the data labelling and clustering results, the device-types are observed to have 99% of the inter-arrival time values. Moreover, similar results are observed for the device-types in the passive network traffic datasets presented in Tables A.31 - A.38. These results show that not all networks have abnormal patterns, and there will be cases where the network has small, large or no abnormal patterns. The run time of the algorithm is between 1.57 and 5.51 seconds.

Table 5.3 Data labelling for iPerf-UDP-Case 1 Passive network traffic dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer	45,000,000	45,000,000	0	0	5.51
Asus	45,000,000	45,000,000	0	0	5.20
Gateway NB	36,000,000	36,000,000	0	0	3.35
Google Phone	15,598,782	15,402,952	195,830	1.3	3.25
Lenovo	15,954,580	15,710,481	244,099	1.5	1.57
Tablet	15,340,910	15,097,390	243,520	1.6	4.53

5.4 Chapter Summary

In summary, this chapter developed a novel device-type profiling approach for smartphones, tablets and laptops using the packet inter-arrival time. The chapter introduced the device-type profiling, profiling algorithm, experimental settings, and data labelling for the device-type profiles. The objectives of this chapter were achieved based on a series of experiments on active, isolated and passive network traffic datasets. The device-type profiling and data labelling experiments were conducted using a clustering-based multivariate gaussian outlier score algorithm whose implementation is available in RapidMiner Studio [173]. Based on the results and analysis of these experiments, the device-type profiling was successful as the normal and abnormal inter-arrival time points for each device-type were observed and profiled.

Regarding the data labelling, we ensured that the correct labels were added to each device-type profile using an expression (if (outlier $\geq$ 1, 'Abnormal', 'Normal')) to count and assign the outlier scores $>$ 1.0 into abnormal or, otherwise, normal profiles, respectively. These expressions were generated based on the standard value defined by [143]. The labels were added to the inter-arrival time points associated with each device-type profile following step 8 of algorithm 5.1. The following chapter of this research uses the device-type labelling results and trains a long short-term memory (LSTM) network to identify, classify and predict these abnormal profiles. The LSTM is selected because of its advantages over other algorithms used for time-series data as well as its dynamic filtering capability. The dynamic filtering capability allows recurrent networks to continue to learn over multiple time steps to identify the abnormal inter-arrival times for each device type.

Chapter 6

Intelligent Filtering Technique using Long Short-Term Memory

An intelligent filtering technique (IFT) was developed using the data labelling results from the device-type profiling presented in chapter 5. It was then implemented using the Bidirectional Long Short-Term Memory architecture of LSTM to identify the abnormal inter-arrival time points from the device-types. The LSTM was proposed by [162] as a solution to the vanishing gradient problem through time and layers. As it maintains more consistent errors, it allows recurrent networks to continue to learn over multiple time steps to identify the abnormal inter-arrival time for each device-type profile. Section 6.1 first introduces the IFT and outlines the experiments and experimental processes used. Then, sections 6.2 and 6.3 focus on the analysis of the IFT training experiments as well as the performance analysis of all the trained IFT for the testing data. Section 6.4 presents a discussion and comparative analysis of this work and the related works. Finally, the chapter is summarised in section 6.5.

6.1 Intelligent Filtering Technique Implementation

The basic idea behind the developed IFT is that the pattern of each device-type's data from the network traffic is trained to identify the abnormal inter-arrival time points, this can be determined based on the training results obtained, and the performance of the trained IFT training are validated on the testing data. The input (i_t) of the IFT is the inter-arrival time values available in each device-type. Then, the output of the IFT (O_t) is the classification of the normal and abnormal inter-arrival time points obtained for each device-type implemented using the bidirectional architecture of LSTM (BiLSTM).

The BiLSTM layer learns bidirectional long-term dependencies between time steps of time series or sequence data. These dependencies are useful because they can give an IFT the ability to learn from the complete time series for each device-type at each time step [162], [178]. The LSTMs and their bidirectional variants are popular because they have the ability to learn how and when to forget and when not to use gates (input, forget and output gates) in their architecture [179], unlike in the previous recurrent neural network architectures where vanishing gradients do not have sufficient learning capability [180].

The advantage of using BiLSTM in IFT is that it consists of two LSTMs that process the inter-arrival times input in a forward direction and reverse direction to identify the abnormal patterns using its three gates (input, forget and output). The reverse direction runs from the past to the future and the forward from future to past. This approach differs from the main LSTM architecture that runs the input unidirectionally [181]. The unidirectional LSTM runs backwards to preserve information from the future and uses the two hidden states combined to preserve information from both past and future. The main significance of using BiLSTM in IFT is that it improves the learning of long-term dependencies and consequently improves the accuracy of the classifier performance. This advantage helps in the identification of abnormal inter-arrival time patterns from device-type profiles since it is time series data. The BiLSTM provides better prediction, more specifically, it was observed that BiLSTM provide better predictions compared to regular unidirectional LSTM [182].

6.1.1 Overview of the IFT Experiments

The experiments are performed in a dedicated lab provided by De Montfort University with the best available, highest specification computers. The lab consists of sixteen HP Eliteone Desktop computers with an Intel (R) Core(TM) i5-7500 CPU at 3.40GHz, and 8.00 GB of RAM. The Operating Systems (OS) of the machines are Windows 10 Education (Version 1709). Also, all the experiments are carried out in MATLAB's 2019a Neural Network Toolbox 11.0 using BiLSTM architecture of the LSTM [178]. The BiLSTM architecture used for the experiments easily handles the problem of abnormal inter-arrival time identification in one-dimensional inter-arrival time sequences. This was validated internally using a testing set presented in section 6.3. The overview of the testing samples used for the internal validation for each device-type is shown in Table 6.1.

Moreover, the overview of the device-types used for training and testing the IFT is presented in Table 6.1. The tables show the total number of the inter-arrival time

values for each device-type in active, isolated, and passive network traffic datasets. The device-types on the table are the only device-types used because they are the most commonly used among active, isolated, and passive network traffic datasets, falling into cases 1, 2 and 3, respectively. This will help to understand how IFT performed in different network traffic measurements and protocols, and whether the IFT can be generalised in different network traffic and protocols. Moreover, other experiments are carried out on the other device-types available in datasets preprocessed in section 5.3, although there are cases where some device-types and datasets are not experimented due to limited computational power and the random access memory's inability to handle large data contained in the device-type profile datasets, for example, the Dell Netbook in isolated network traffic, among others. The results of the other experiments that are not presented here are similar to those reported in section 6.3, therefore, reporting additional experiment results would lead to long and tedious analysis without any significant added value.

Table 6.1 An overview of all the Network Traffic Training and Testing Samples used for the implementation of IFT.

		Real Datasets Training and Testing Split Ratio	
Network Traffic Type	Device-Type	Training IATs (80%)	Testing IATs (20%)
Active	Acer NB	3,174,825	793,718
	Asus NB	3,174,850	793,975
	Gateway NB	2,543,935	635,996
	Google Phone	637,405	159,363
	Lenovo Laptop	637,613	159,662
	Asus Tablet	635,931	158,995
Isolated	iPad	3,655,182	916,308
	iPhone 3G	903,479	225,870
	iPhone 4G	6,640,562	1,660,153
	Nokia	1,250,360	312,602
Passive	Acer NB	2,569,901	642,487
	Asus NB	2,569,713	642,440
	Gateway NB	2,055,754	513,951
	Google Phone	520,123	130,043
	Lenovo Laptop	514,021	128,517
	Asus Tablet	513,953	128,500

Moreover, recalling from section 5.3, all the device-types are observed with approximately between $0.1 - 3.1\%$ abnormal inter-arrival time points, with other points considered normal, without adequate data preprocessing such as normalisation, and stratification steps presented in Algorithm 6.1, the LSTM will not converge towards high accuracy because the datasets have only a small number of abnormal inter-arrival

time points. The data preprocessing steps are described in the experimental process section.

6.1.2 IFT Experimental Process

The normalisation is the most important part in ensuring that all input sequences are similar in terms of value range. A simple "division by maximum value" data normalisation was used for each individual sequence, so each abnormal inter-arrival time point is presented in the local context of the given sequence. Another parameter in the data preparation step for training is "ratioNegativeToPositive". This handles the problem of highly imbalanced datasets like the one used in our experiments (there are approximately between $0.1 - 3.1\%$ abnormal inter-arrival time points in all the device-types, with other points considered normal). In the first training (10 epochs) all abnormal samples from the training set are taken along with double the number of normal samples, which is a 33% to 66% distribution between the two classes. The IFT was trained to an accuracy of $> 99\%$. After that the IFT was retrained but with a ratioNegativeToPositive = 10, meaning that now we have an approximately 9% to 91% distribution between the two classes, which is closer to the real case of 3% versus 97% in the class distribution. Without this second training the classifier would be too biased towards the positive class and would label many normal sequences as abnormal.

As for the training and testing parameter settings for the IFT, the steps in Algorithm 6.1 is followed accordingly. Algorithm 6.1 shows the data preprocessing processes that includes inputting the data, selecting the number of sequences and hidden layers, and splitting the data into training and testing sets. As well as the training options for BiLSTM, how the training was performed, and visualises the outputs for training and testing for each device-type. Moreover, the algorithms show the parameters used in the experiments for inputting training and testing samples, and the LSTM parameter configurations. The goal of data input is to load the data for each device-type into the workspace and split into training and testing set. Whereas, the LSTM parameter configuration shows the best settings used in configuring the LSTM for training and testing the IFT.

For the input parameters, the data for each device type was experimented on using different training and testing split ratios, e.g., starting from 60/40 increasing to 70/30 and 80/20 and comparing the performance results; they all worked well. However, 80/20 gave the best performance with low false detection rates; therefore, 80% for training and 20% for testing split ratio were chosen as the best input parameters for all the experiments. Another reason for choosing 80% for training and 20% for testing

was because it is always better to have as much data in the training set as possible while still keeping enough in the testing set for the testing to be valid/strong enough [183]. The LSTM parameters for training and testing are sequence length, number of hidden layers, and the training options. The sequence length is important because the abnormal versus normal inter-arrival time points for each device-type have to be distinguishable in every given sequence, and this is in itself highly dependent on the sequence length. A length 50 was used and indeed appeared to work well, though it is likely that slightly shorter or longer sequences would work too (but not too short). Lastly, all the steps shown in the algorithms are the best parameters that gave the best performance among all the other settings we explore.

Algorithm 6.1 Device-Type IFT algorithm

```
1  procedure preprocessing, normalisation and stratification
       steps
2  LOAD device-type IAT data into MATLAB workspace
3  Preprocess the data into a sequence of double arrays
4  Preprocess the target variable as a categorical array
5  Training and Testing steps for BiLSTM
6        SET the sequence size to 50
7        Randomise and split the data into 80% for training and
           20% for testing
8        Organise the sequences into a cell array
9        SET bidirectional LSTM layer to value 50
10       Apply softmax function to the input in the softmax layer
11 Classification
12       Compute cross-entropy loss
13       SET training algorithm to adaptive moment estimation (
           ADAM) solver.
14       SET Max Epochs to 10
15       SET Sequence Length parameter to 'longest'
16       SET Gradient Threshold to 1
17 Fixing imbalance problem for IFT classification
18       Take all N positive samples from training set and 2 x N
           randomly selected negative samples.
19       SET Epochs to 10
20       Take all N positive samples from training set and 10 x N
           randomly selected negative samples
21       SET Epochs to 5
```

```
22  Display training progress as number of iterations increases
23  Display training and testing accuracy in the performance
       confusion matrix.
24  End procedure
```

6.2 Analysis of the IFT Training Results

The inter-arrival time values used for training the IFT for all the device-types in the active, isolated and passive network traffic datasets is illustrated in Table 6.1. From the table, it can be seen that 80% of the inter-arrival time points for each device-type was used for training the IFT so that the parameter estimate will not have a great variance with less training data. Also, the IFT results using 80% training samples gave an accuracy $> 95\%$ for most of the device-types in active, isolated and passive network traffic datasets. The training results are analysed in the following sections and the training states for all the experiments is presented in Figures B.2 - B.32 in Appendix B. The training state show the training accuracy and errors during training the IFT, the training time for each device-type, and the training cycles along with the number of iterations. The IFT training results is presented so that when the IFT is tested we can compare the training and testing results to observed any high bias and high variance in the IFT. The high bias happens when there is under fitting i.e., the IFT is not presenting an accurate picture of the relationship between the inputs and predicted output. Whereas, the high variance happens when there is overfitting which is the opposite of high bias, meaning the IFT work well on the training sets but will not know how well it performed until compared the training against the test results. Therefore, next section present the analysis of the IFT training results and the following section evaluates the IFT on testing sets.

6.2.1 Analysis of the Active Network Traffic Dataset

This network traffic dataset consists of inter-arrival time data points for Acer, Asus, Gateway and Lenovo as well as one Asus Tablet and a Google Phone in all the two active network traffic datasets, as stated in section 4.1.2. The experiment was conducted on all the datasets, however, only the results of all device-types from one dataset are presented in this section because there were no significant differences between the results for the chosen device-types and the other device-types in the other datasets. A full consideration of all the datasets would lead to a long and tedious analysis without

significant added value. However, in cases where there are significant differences in the results, all the datasets are considered in the analysis. Figures 6.1 - 6.6 present the IFT training results for the device-types in the chosen datasets. Here, the diagonal cells shaded green show the number of correctly classified inter-arrival time points, while the cells shaded red show the number of incorrectly classified inter-arrival time points.

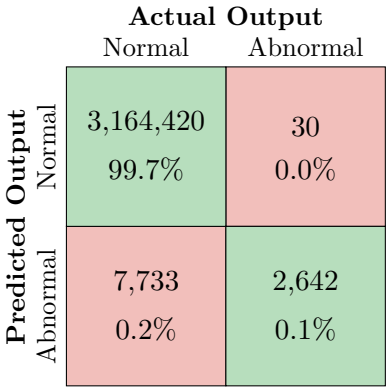


Fig. 6.1 The intelligent filtering technique training confusion matrix for Acer Netbook in active network traffic dataset.

Acer Netbook

From the actual output of the IFT training confusion matrix for the Acer Netbook presented in Figure 6.1, it was observed that the IFT correctly identified 3,164,420 and 2,642 falling into normal and abnormal profiles, respectively. For the predicted output, the IFT incorrectly predicts 7,733 inter-arrival time points falling into the normal profile. Also, predicts 30 inter-arrival time points falling into the abnormal profile. Moreover, from the diagonal cells shaded green the total inter-arrival time points that are correctly identified by the IFT is 3,167,062, which corresponds to 99.9% of the training samples. Similarly, from the diagonal cells shaded red, 7,763 is the total inter-arrival time points that were incorrectly identified or the errors made by the IFT during training, which corresponds to 0.1% of the training samples.

This training result show that the IFT correctly identify the abnormal inter-arrival time points from this device-type, however, to measure the efficiency and effectiveness of the IFT, it will be evaluated on the testing sets. The performance on the testing sets should be able to judge whether the IFT outperformed or not. The way it shows whether it outperformed or not is when the test results return higher true negative rates and lower false positive rates [184], [185], similar measure apply to all the device-types trained and analyse in this section.

Asus Netbook

From the actual output of the IFT training confusion matrix for the Asus Netbook presented in Figure 6.2, it was observed that the IFT correctly identified 3,171,687 and 1,515 inter-arrival time points falling into normal and abnormal profiles, respectively.

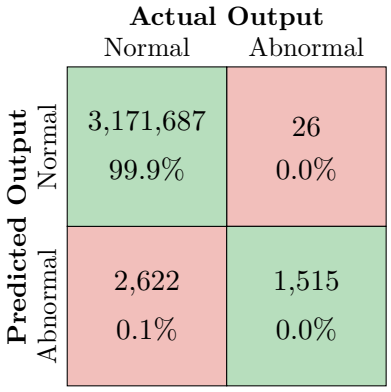


Fig. 6.2 The intelligent filtering technique training confusion matrix for Asus Netbook in active network traffic dataset.

In the predicted output, the IFT predicts 26 inter-arrival time points falling into the normal profile and 2,622 inter-arrival time points falling into the abnormal profile. Meanwhile, from the overall training samples for this device-type, the total inter-arrival time points that are correctly identified is 3,173,202 and incorrectly identified is 2,648, these values are calculated by calculating the summations of the total inter-arrival time points in the diagonal cells shaded green and red. In this device-type, it was observed that there is only 0.1% error in the IFT training.

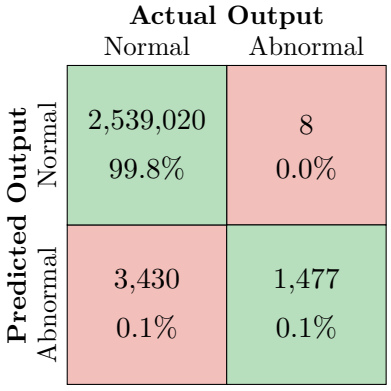


Fig. 6.3 The intelligent filtering technique training confusion matrix for Gateway Netbook in active network traffic dataset.

Gateway Netbook

The training confusion matrix of the IFT for the Gateway Netbook is presented in Figure 6.3. In the actual output, it was observed that the IFT correctly identified 2,539,020 inter-arrival time points in the normal profile, while incorrectly identifying

1,477 of the inter-arrival time points in the abnormal profile. Besides, in the predicted output, the IFT predicts 8 inter arrival time points for normal profile and 3,430 for abnormal profile. Overall, the total inter-arrival time points used for training this device-type is 2,543,935, in which the IFT correctly identified 2,540,497 inter-arrival time points and incorrectly identifying 3,438 inter-arrival time points, which indicated that there is only 0.1% error in the IFT training.

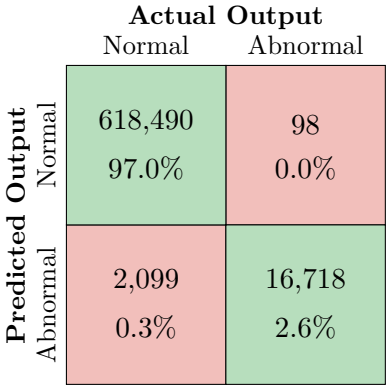


Fig. 6.4 The intelligent filtering technique training confusion matrix for Google Phone in active network traffic dataset.

Google Phone

The IFT training confusion matrix for the Google Phone is presented in Figure 6.4. In the actual output, the IFT correctly identified 618,490 inter-arrival time points in the normal profile and 16,718 inter-arrival time points falling into the abnormal profile. Moreover, the incorrectly predicted inter-arrival time points in the normal and abnormal profiles are 98 and 2,099 with the total points corresponding to 0.3% of the training samples. Similarly, the inter-arrival time points in diagonal cells represents the total inter-arrival time points that are correctly and incorrectly trained by the IFT which is 635,208 in the cells shaded green, and 2,197 in the cells shaded red. This device-type was observed with 0.3% error during the IFT training.

Lenovo Laptop

The IFT training confusion matrix for the Lenovo Laptop is presented in Figure 6.5. In the actual output, it correctly identified 618,559 which is 96.9% of the inter-arrival time points falling into the normal profile and 18,766 corresponding to 2.9% into the abnormal profile. The incorrectly predicted inter-arrival time points in the normal

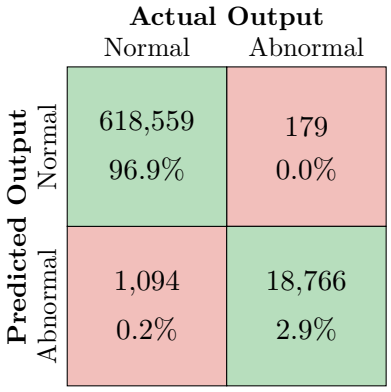


Fig. 6.5 The intelligent filtering technique training confusion matrix for Lenovo Laptop in active network traffic dataset.

profile observed was 179 and 1,094 for abnormal profile corresponding to 0.3% of the training samples. Moreover, from the diagonal cells shaded green and red, the IFT correctly identified 99.7% and incorrectly identified 0.3%, which corresponds to 637,325 and 1,273 of the inter-arrival time points in training. Like in the other device-types analysed above this device-type was observed with only 0.3% error during the IFT training.

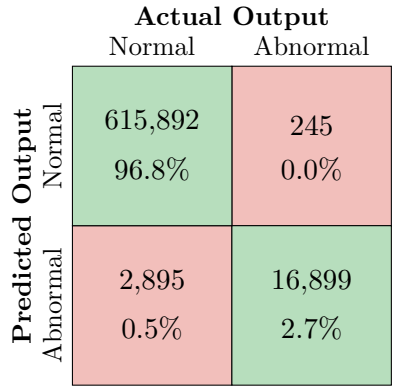


Fig. 6.6 The intelligent filtering technique training confusion matrix for Asus Tablet in Active active network traffic dataset.

Asus Tablet

The IFT training confusion matrix for the Asus Tablet is presented in Figure 6.6. In the actual output, the IFT correctly identified 615,892 and 16,899 inter-arrival time points corresponding to 96.8% and 2.7% of the inter-arrival time points falling into the normal and abnormal profiles, respectively. The incorrectly predicted inter-arrival

time points in the normal profile observed was 245 and for abnormal profile 2,895 corresponding to 0.5% of the training samples. Moreover, from the diagonal cells shaded green and red, the IFT correctly identified 632,798 corresponding to 99.5% and incorrectly identified 3,140 corresponding to 0.5% of the inter-arrival time points in training. The overall error made by the IFT in training this device-type is 0.5%.

6.2.2 Analysis of the Isolated Network Traffic Dataset

This network traffic dataset consists of inter-arrival time values for the Dell Netbook, iPad, iPhone 3G, iPhone 4G, and Nokia Phone. In this case, we observe that the IFT was able to identify the abnormal inter-arrival time points for all the device-types; however, the experiment was not conducted for the Dell Netbook due to the large number of inter-arrival time points it contained, which could not be handled by the computers available for use in these experiments. One of our findings is that the IFT results for most of the device-types in this network traffic dataset are similar to those for the device-types analysed in section 6.2.1. In addition, we experimented with the other six datasets in the isolated network traffic dataset using similar device-types and observed similar performances. Since the IFT performance is good in this network traffic datasets, we analysed the device-types from TCP network traffic to facilitate an analysis that was different from that of active and passive network traffic traces.

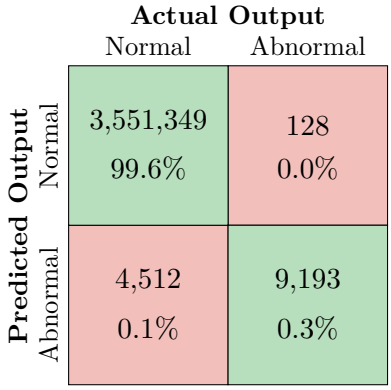


Fig. 6.7 The intelligent filtering technique training confusion matrix for iPad in isolated network traffic dataset.

iPad

The IFT training confusion matrix for the iPad is presented in Figure 6.7. For this device-type, the IFT correctly identified 3,651,349 inter-arrival time points for the

normal profile and 9,193 for the abnormal profile in the actual output class. In the predicted output, the IFT incorrectly predicted 128 inter-arrival times for the normal profile and 4,512 for abnormal profile. Meanwhile, the total inter-arrival time points calculated from the diagonal cells shaded green and red, it was observed that the IFT correctly identified 3,660,542 of the inter-arrival time points and incorrectly identified 4,640 from the training samples, which indicated that the IFT had 0.1% error in training this device-type.

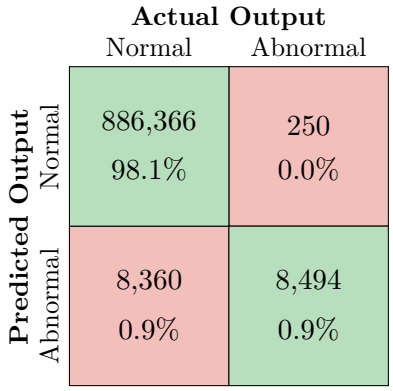


Fig. 6.8 The intelligent filtering technique training confusion matrix for iPhone 3G in isolated network traffic dataset.

iPhone 3G

The IFT training confusion matrix for the iPhone 3G is presented in Figure 6.8. From the actual output, it can be seen that 886,366 and 8,494 inter-arrival time points are correctly identified falling into normal and abnormal profiles. Meanwhile, the predicted inter-arrival time points for the normal profile is 250 and 8,360 for abnormal profiles, respectively. Moreover, from the total inter-arrival time points in the diagonal cells shaded green and red, the IFT correctly identified 894,860 and incorrectly identified 8,610 inter-arrival time points from the training sets. Like in the other device-types analysed above this device-type was observed with 0.9% error during the IFT training.

iPhone 4G

The IFT training confusion matrix for the iPhone 4G is presented in Figure 6.9. As can be seen from the figure, the actual output the IFT correctly identified 6,519,825 inter-arrival time points in the normal profile and 33,516 inter-arrival time points falling into the abnormal profile. Moreover, the incorrectly predicted inter-arrival time

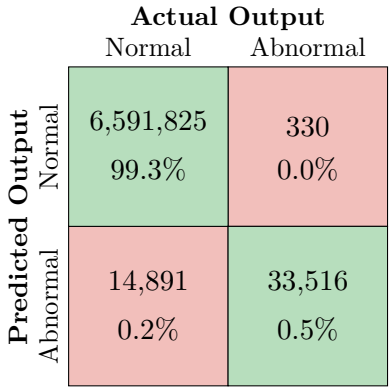


Fig. 6.9 The intelligent filtering technique training confusion matrix for iPhone 4G in isolated network traffic dataset.

points in the normal and abnormal profiles are 330 and 14,891 inter-arrival time points. Whilst, from the total inter-arrival time points in the diagonal cells shaded green and red the IFT was observed to have correctly identified 6,591,825 of inter-arrival time points and incorrectly identified 15,221 in the training. The overall error made by the IFT in training this device-type is 0.2%.

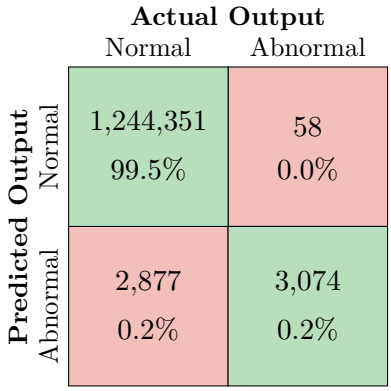


Fig. 6.10 The intelligent filtering technique training confusion matrix for Nokia Phone in isolated network traffic dataset.

Nokia Phone

The IFT training confusion matrix for the Nokia Phone is presented in Figure 6.10. In the actual output, it correctly identified 618,559 which is 99.5% of the inter-arrival time points falling into the normal profile and 18,766 corresponding to 0.2% into the abnormal profile. Moreover, in the predicted output class, the normal profile was observed with 58 inter-arrival time points and abnormal profile with 2,877 corresponding

to 0.2% of the training samples. Moreover, the total inter-arrival time points observed in the diagonal cells shaded green and red, the IFT correctly identified $1,247,424$ and incorrectly identified $2,935$ of the inter-arrival time points in training. Overall, this device-type was observed with 0.2% error during the IFT training.

6.2.3 Analysis of the Passive Network Traffic Dataset

This network traffic dataset consists of network traffic traces for the six device-types in the UDP traffic of the passive network traffic dataset. The six device-types are the Acer, Asus, and Gateway Netbooks, Google Phone, Lenovo Laptop, and Asus Tablet. For these device-types, we observe that the intelligent filtering technique correctly identified the majority of the abnormal inter-arrival time points. Furthermore, the results demonstrate a similar performance with the other device-types in both the active and isolated network traffic datasets. Moreover, there are seven different datasets in the passive network traffic dataset, whereby experiments were conducted for similar device-types and we observe performances similar to what is presented in the analysis below. Since the identification by the IFT is good in these network traffic datasets, we analysed UDP traffic so that we could conduct an analysis different from those performed for the active and isolated network traffic datasets. This will help in the evaluation chapter, where the results for the isolated, active and passive network traffic datasets are compared. However, this section only analyses the UDP network traffic dataset, since the TCP and ICMP are analysed in the previous sections above.

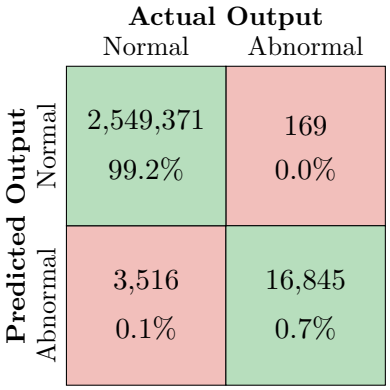


Fig. 6.11 The intelligent filtering technique training confusion matrix for Acer Netbook in passive network traffic dataset.

Acer Netbook

The IFT training confusion matrix for the Acer Netbook is presented in Figure 6.11. In the actual output, the IFT correctly identified 2,549,371 and 16,845 inter-arrival time points in the normal and abnormal profiles, respectively. In the predicted output, the IFT incorrectly identified 169 inter-arrival time points being normal and incorrectly identified 3,516 as abnormal from the training samples. Moreover, the total inter-arrival time points observed in the diagonal cells shaded green and red, the IFT correctly identified 2,566,216 and incorrectly identified 2,685 of the inter-arrival time points in training. Overall, this device-type was observed with 0.1% error during the IFT training.

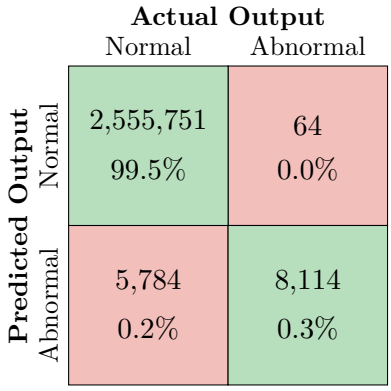


Fig. 6.12 The intelligent filtering technique training confusion matrix for Asus Netbook in passive network traffic dataset.

Asus Netbook

From the actual output of the IFT training confusion matrix for the Asus Netbook presented in Figure 6.12, it can be seen that 2,555,751 and 8,114 inter-arrival time points are correctly identified in the normal and abnormal profiles. Meanwhile, the incorrectly predicted inter-arrival points for the normal profile is 64 and 5,784 for abnormal profile. Meanwhile, from the overall training samples for this device-type, the total inter-arrival time points that are correctly identified is 2,563,865 and incorrectly identified is 5,848, these values are calculated by calculating the summations of the total inter-arrival time points in the diagonal cells shaded green and red. In this device-type, it was observed that there is only 0.2% error in the IFT training.

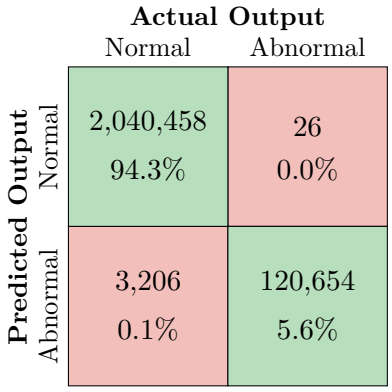


Fig. 6.13 The intelligent filtering technique training confusion matrix for Gateway Netbook in passive network traffic dataset.

Gateway Netbooks

The Gateway Netbook confusion matrix for the training is presented in Figure 6.13. For the actual output this device-type, it was observed that the IFT correctly identified 2,040,458 for normal profile and 120,654 for abnormal profile. Meanwhile, in the predicted output, the inter-arrival time points observed in the normal profile is 26 and 3,206 in the abnormal profile. Overall, the total inter-arrival time points used for training this device-type is 2,055,754, in which the IFT correctly identified 2,161,112, while incorrectly identifying 3,232 of the inter-arrival time points in the training samples, which indicated that there is only 0.2% error in the IFT training.

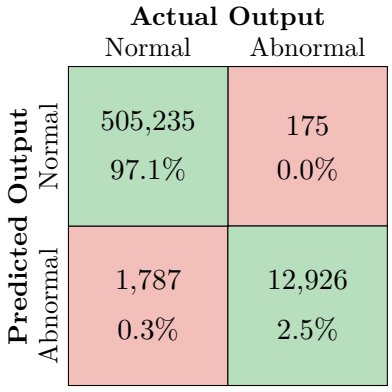


Fig. 6.14 The intelligent filtering technique training confusion matrix for Google Phone in passive network traffic dataset.

Google Phone

The IFT training confusion matrix for the Google Phone is presented in Figure 6.14. In the actual output the IFT correctly identified 505,235 inter-arrival time points in the normal profile and 12,926 inter-arrival time points falling into the abnormal profile. Moreover, the incorrectly predicted inter-arrival time points in the normal and abnormal profiles are 175 and 1,787 inter-arrival time points, respectively. Similarly, the inter-arrival time points in diagonal cells represents the total inter-arrival time points that are correctly and incorrectly trained by the IFT which is 518,161 inter-arrival time points in the cells shaded green, and 1,962 in the cells shaded red. Overall, this device-type was observed with 0.3% error during the IFT training.

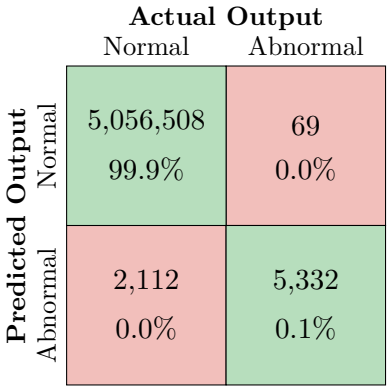


Fig. 6.15 The intelligent filtering technique training confusion matrix for Lenovo Laptop in passive network traffic dataset.

Lenovo

The IFT confusion matrix for the Lenovo Laptop is presented in Figure 6.15. In the actual output, it shows that the IFT correctly identified 5,056,508 and 5,332 inter-arrival time points in both the normal and abnormal profiles, respectively. In the predicted output, the IFT predicted 69 and 2,112 inter-arrival time points in the normal profile abnormal profiles, respectively. Moreover, from the diagonal cells shaded green and red, the IFT correctly identified 5,061,840 and incorrectly identified 2,181 inter-arrival time points from the training samples. Like in the other device-types analysed above this device-type was observed with 0.4% error during the IFT training.

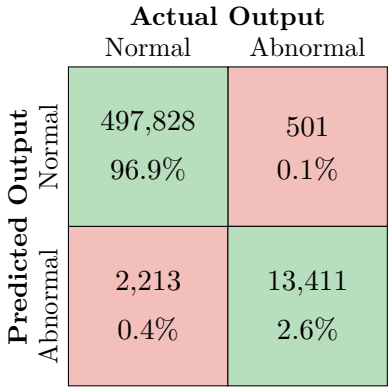


Fig. 6.16 The intelligent filtering technique training confusion matrix for Asus Tablet in passive network traffic dataset.

Asus Tablet

The IFT confusion matrix for the Asus Tablet is presented in Figure 6.16, from the actual output, the IFT correctly identified 497,828 and 13,411 inter-arrival time points in the normal and abnormal profiles, respectively. In the predicted output, the IFT predicts 501 and 2,213 inter-arrival time points in both normal and abnormal profiles. Moreover, from the diagonal cells shaded green and red, the IFT correctly identified 511,239 and incorrectly identified 2,714 inter-arrival time points in training. The overall error made by the IFT in training this device-type is 0.5%.

6.3 Performance Evaluation of the IFT

The summary of the remaining 20% testing sets illustrated in Table 6.2 are used to evaluate the performance of the IFT in the unseen data and to calculate the prediction error. The prediction error is calculated based on the misclassification error metric, which gives a binary output by simply testing whether each prediction is correct or incorrect. This binary output is used to calculate the true positives and true negatives (the percentage of inter-arrival time points correctly identified by the IFT) and false positives and false negatives (the percentage of inter-arrival time points incorrectly identified by the IFT). Moreover, as the IFT is a two-class problem in which the classifier decides over a set of objects (i.e., normal and abnormal inter-arrival time points), binary evaluation metrics are used in IFT evaluation. These metrics are accuracy, recall (also known as sensitivity or true positive rate), precision (positive predictive value) and F-score, and they provide quantifiable evidence as to how effective the IFT is at making correct predictions.

Table 6.2 An overview of all the Network Traffic Testing Samples used for the implementation of IFT.

		Real Datasets Testing Samples
Network Traffic Type	Device-Type	Testing IATs (20%)
Active	Acer NB	793,718
	Asus NB	793,975
	Gateway NB	635,996
	Google Phone	159,363
	Lenovo Laptop	159,662
	Asus Tablet	158,995
Isolated	iPad	916,308
	iPhone 3G	225,870
	iPhone 4G	1,660,153
	Nokia	312,602
Passive	Acer NB	642,487
	Asus NB	642,440
	Gateway NB	513,951
	Google Phone	130,043
	Lenovo Laptop	128,517
	Asus Tablet	128,500

Other complementary measures are Specificity (SPC), the Negative Predictive Value (NPV), and False Positive Rates (FPR). These are used in highly imbalanced classification tasks, such as anomaly detection, where a biased classifier achieves high recall (sensitivity) at the expense of low precision. This behaviour is acceptable when the system requirements strongly demand a certain action in the case of an anomaly, but that same action is not as critical in the case of a false positive. An obvious example would be a cancer diagnosis system, where additional check-ups are not harmful in the case of a false-positive diagnosis yet can save a patient's life in the case of a true positive. Hence, very good recall is essential, and relatively bad precision is acceptable. This justifies the effectiveness of the IFT in all the device-types in cases with low precision. The respective metric equations are defined below:

- The Accuracy is defined as the number of correctly classified inter-arrival time points out of all the inter-arrival time points, represented as:

$$\frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \times 100 \tag{6.1}$$

  where TP are the true positives, TN the true negatives, FP the false positives and FN are the false negatives. Also, the TP are the inter-arrival time points

that are correctly classified as the actual class(es), TN are the inter-arrival time points that are correctly classified as not being the actual class(es), FP is the error, or inter-arrival time points that are incorrectly classified as the actual class, and the FP are the error inter-arrival time points from the actual class that are incorrectly classified as (an)other class(es).

- Precision is defined as the percentage of positive inter-arrival time points within all positive labelled inter-arrival time points, represented as:

$$\frac{\text{TP}}{\text{TP} + \text{FP}} \times 100 \tag{6.2}$$

- Recall is defined as the fraction of correctly classified inter-arrival time points of a particular class within all inter-arrival time points that belong to that class, represented as:

$$\frac{\text{TP}}{\text{TP} + \text{FN}} \times 100 \tag{6.3}$$

- F-Score is defined as the weighted harmonic mean of precision and recall represented as:

$$\frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \times 100 \tag{6.4}$$

- Specificity (SPC) is defined as the number of correct negative predictions divided by the total number of all negatives. The best specificity is 1.0, and the worst is 0.0. The recall is represented as:

$$\frac{\text{TN}}{\text{TN} + \text{FP}} \times 100 \tag{6.5}$$

- Negative Predictive Value (NPV) is defined as the number of incorrect negative predictions divided by the total number of all negatives and the best NPV is 1.0 whereas the worst is 0.0. The negative predictive value is represented as:

$$\frac{\text{TP}}{\text{TN} + \text{FN}} \times 100 \tag{6.6}$$

- False Positive Rate (FPR) is defined as the number of incorrect positive predictions divided by the total number of all negatives and the best false positive rate is 0.0 whereas the worst is 1.0 and it is represented as:

$$\frac{\text{FP}}{\text{TN} + \text{FP}} \times 100 \tag{6.7}$$

6.3.1 Evaluation of Device-Types in Active Network Traffic

The performance evaluation of the testing sets for the Acer, Asus and Gateway Netbooks, one Lenovo Laptop as well as the Google Phone and Asus Tablet is presented below. The performance evaluation was based on the testing samples illustrated in Table 6.1.

Table 6.3 The Evaluation Metrics For the Device-Types in Active network traffic datasets

Device	**Accuracy** (%)	**Precision** (%)	**Recall** (%)	**F-Score** (%)	**SPC** (%)	**NPV** (%)	**FPR** (%)
Acer	99.9	100	99.9	99.9	98.3	41.5	2
Asus	99.9	100	99.9	99.9	99.7	24.4	0.3
GatewayNB	99.9	100	99.9	99.9	97.8	37.7	2
GoogleP	99.6	100	99.6	99.8	99.2	88.9	0.8
Lenovo	99.9	100	99.9	99.9	99.5	94.2	0.5
Asus Tablet	99.3	99.9	99.4	99.6	98.2	85.6	2

The performance evaluation of the studied device-types in the active network traffic datasets is presented in Table 6.3. The table shows the device-types and the different metrics used to measure the IFT's effectiveness and efficiency in identifying abnormal inter-arrival time points. Hereby, the table demonstrates that the accuracies for the device-types are 99.3, 99.6, and 99.9% and the recall for the device-types is 99.9%, except for the Google Phone and Asus Tablet, which have 99.6 and 99.4%, respectively. Additionally, the Precision for the device-types is 100%, except for the Asus Tablet, for which it is 99.9%, and the F-score falls between 99.6 and 99.9%. For example, regarding the interpretation of these metrics in the Acer Netbook, the inter-arrival time points used for testing the IFT is 793,718; out of these inter-arrival time points, the IFT correctly identified 99.9% and incorrectly identified 0.1%, corresponding to 792,764 and 954 inter-arrival time points, respectively. As for the recall, the IFT identified 99.9% of the inter-arrival time points, corresponding to 793,409 inter-arrival points in the normal profile, while identifying 0.1%, corresponding to 309 inter-arrival time points, in the abnormal profile. Regarding the precision (predicted output), the IFT predicted 99.8% and 0.2% of the inter-arrival time points (i.e. 792,457 and 1,261) in the normal and abnormal profiles, respectively, meaning the total precision is 100%.

Moreover, the accuracy, precision, recall, and F-score are higher (i.e. 99.9%), which shows that the IFT outperforms on the testing sets. It should be noted that as the datasets are imbalanced, with most of the inter-arrival time points being labelled as normal, achieving high classification accuracy is not difficult. To ensure the correctness of the IFT identification, the performance is further justified using SPC, NPV, and FPR. The SPC was used to measure the efficiency of the IFT in correctly identifying the abnormal inter-arrival time points, whereas the NPV and FPR measured the effectiveness of the IFT identification of the abnormal inter-arrival time points. The SPC falls between 97.8 and 99.7%, whereby the Gateway Netbook has the lowest SPC of 97.8% and the Asus Netbook has the highest SPC at 99.7%. In contrast, the NPV ranges between 24.4 and 94.2%, with the Asus Netbook having the lowest NPV and the Lenovo Laptop having the highest NPV, and the FPR ranges between 0.3 and 2%. These metrics show that the intelligent filter is not biased towards false positive or false negative classes but correctly identifies both normal and abnormal inter-arrival time points. Hence, according to the NPV, the Lenovo Laptop had the best performance, followed by the Google Phone and the Asus Tablet, then the Acer and Gateway Netbooks as well as the Asus Netbook, which has the lowest NPV among the device-types, this low NPV was because of the fewer false negatives. Hence, the FPR is below 1% and the SPC is greater than 80%, therefore the performance is good as it has reached the acceptable standard for a workable intrusion detection system [186], [187], [188], meaning that the NPV will not affect the IFT identification. Moreover, the network administrators will be able to identify intrusions when there is a decrease in the NPV and the false alarms when there is an increase in the FPR.

6.3.2 Evaluation of Device-Types in Isolated Network Traffic

The performance evaluation of the testing sets for the iPad, iPhone 3G, iPhone 4G, and Nokia Phone is presented below. The performance evaluation is based on the testing samples illustrated in Table 6.1.

Table 6.4 Testing Evaluation Metrics For Isolated network traffic datasets

Device	**Accuracy** (%)	**Precision** (%)	**Recall** (%)	**F-Score** (%)	**SPC** (%)	**NPV** (%)	**FPR** (%)
iPads	99.9	100	99.9	99.9	99.1	89.5	0.9
iPhone 3G	98.2	99.8	98.4	99.0	93.3	64.8	7
iPhone 4G	99.8	100	99.8	99.9	99.4	69.8	0.6
Nokia	99.9	100	99.9	99.9	98.6	70.7	1

The performance evaluation of the studied device-types in the isolated network traffic datasets is presented in Table 6.4. From the table, we observe that the accuracies of the four device-types lie between 98.2 and 99.9%, the recall is between 98.4 and 99.9%, and the precision is 100%, except for the iPhone 3G, which has 99.8%. Besides, the F-score for the device-types is 99.9%, except for the iPhone 3G, which has 99.0%. As an example, in the interpretation of these metrics for the iPad, the number of inter-arrival time points used for testing the IFT is 916,308; out of these, the IFT correctly classified 916,049 and incorrectly classified 259, corresponding to 99.9% and 0.1% of the inter-arrival time points, respectively. As for the actual output, the IFT identified 914,253 inter-arrival time points, corresponding to 99.9% of the inter-arrival points in the normal profile, while identifying 2,036, corresponding to 0.1%, of the inter-arrival time points in the abnormal profile. Moreover, in the predicted output, the IFT predicted 914,032 and 2,276 of the inter-arrival time points in the normal and abnormal profiles, respectively. A similar performance was observed for the iPhone 4G and Nokia Phone, although each device-type had a different number of inter-arrival time points.

Moreover, the accuracy, precision, recall, and F-score are higher (i.e. 99.9%), showing that the IFT outperformed on the testing sets. As mentioned above for the active network traffic, we must be cautious with such high accuracies as the dataset is imbalanced and thus accuracy by itself is not a suitable performance metric for this device-type. Therefore, the IFT performance is further justified using the SPC, NPV, and FPR. The SPC falls between 98.6 and 99.4%, whereby the iPhone 4G has the highest (97.8%) and the Nokia Phone has the lowest (99.7%), and the NPV ranges between 64.8 and 89.5%, with the iPad having the highest (89.5%) and the iPhone 3G having the lowest (64.8%). As for the FPR metrics, the values range between 0.6 and 7%, which means that the intelligent filter is not biased towards false positive or false negative classes but instead correctly identifies both the normal and abnormal inter-arrival time points for all the device-type profiles. It should be noted that the iPhone 3G has the largest FPR of all the experimented device-types in the active, and isolated network traffic datasets; this does not affect its performance. Also, the NPV and FPR for all the device-types are below 10% and the NPV is above 60%, which shows that the IFT identification performance for the device-types in the isolated network traffic datasets is slightly better than in the active network traffic datasets. Therefore, the network administrators will be able to identify intrusions when there is a decrease in the NPV and the false alarms when there is an increase in the FPR.

6.3.3 Evaluation of Device-Types in Passive Network Traffic

The performance evaluation of the testing sets for the Acer, Asus and Gateway Netbooks, one Lenovo Laptop as well as the Google Phone and Asus Tablet. The performance evaluation was based on the testing samples illustrated in Table 6.1.

Table 6.5 Testing Evaluation Metrics For Passive network traffic datasets

Device	**Accuracy** (%)	**Precision** (%)	**Recall** (%)	**F-Score** (%)	**SPC** (%)	**NPV** (%)	**FPR** (%)
Acer	99.9	100	99.9	99.9	99.2	87.3	0.8
Asus	99.8	100	99.8	99.9	99.6	53.9	0.4
GatewayNB	99.9	100	99.9	99.9	99.9	77.0	0.08
GoogleP	99.3	99.9	99.3	99.6	95.0	76.4	15
Lenovo	99.6	100	99.6	99.8	99.0	71.2	1
Asus Tablet	99.9	99.9	99.9	99.0	98.3	87.4	3

The performance evaluation of the studied device-types in the active network traffic datasets is presented in Table 6.5. The table demonstrates that the accuracy for the device-types is 99.3, 99.6, and 99.8 - 100%, the recall and F-score are also similar to the accuracy, and the precision falls between 99.9 and 100%. In the Google Phone, for example, we observe that the IFT correctly classifies 129,074 and incorrectly identifies 969 inter-arrival time points out of the total 130,043 used for testing the IFT. As for the actual output (recall), the IFT identified 99.3% of the inter-arrival time points in the normal profile and 0.7% of the inter-arrival time points in the abnormal profile. Moreover, for the predicted output (precision), the IFT predicted 99.9% and 0.1% of the inter-arrival time points in the normal and abnormal profiles, respectively.

Furthermore, the accuracy, precision, recall, and F-score are higher, demonstrating that the IFT outperformed on the testing sets. We reiterate here that such high accuracies may not offer much meaning due to the imbalanced dataset; hence, we use additional performance metrics, such as the SPC, NPV, and FPR. The SPC falls between 98.2 and 99.9%, in which the Asus Tablet has the lowest SPC of 97.8% and Gateway Netbook has the highest SPC of 99.9%. The NPV ranges between 53.8 and 87.4%, with the Asus Netbook having the lowest NPV and the Acer and Asus Tablets having the highest NPV. Hence, based on the NPV, the Acer Netbook and Asus Tablet have the best performance, followed by the Gateway Netbook, Google Phone and Lenovo Laptop, while the Asus Netbook had the lowest NPV among the device-types. As for the FPR, the metrics fall between 0.08 and 15%, which shows that there are more false-positive alerts. In comparison to the active and isolated network traffic datasets, it can be seen that the Google Phone has the largest FPR (15%), although this does not

affect the IFT performance because the FPR was affected by the fewer false negatives and higher SPC as evident from the NPV. These metrics show that the intelligent filter is not biased towards false positive or false negative classes but correctly identifies both the normal and abnormal inter-arrival time points. Furthermore, the NPV and FPR for all the device-types are below 10% and the NPV is above 70%, except in the case of the Asus Netbook, which has 53.9%. These performance metrics show that the IFT identification for the device-types in the passive network traffic datasets is slightly better than in the active and isolated network traffic datasets. Therefore, the network administrators will also be able to identify intrusions when there is a decrease in the NPV and false alarms when there is an increase in the FPR.

6.4 Discussion and Comparison

As previously discussed in the literature review, most of the related works developed fingerprinting and behaviour profiling techniques to address network access control issues. In their approaches, they considered using important attributes, such as devices, device types, access points, and service use, to fingerprint or profile network devices. Contrarily, the proposed work utilizes the important attributes used in fingerprinting and behaviour profiling techniques, e.g., devices, device types, application, location and packet inter-arrival times, to profile and identify abnormal patterns based on device type. The most closely related works among those reviewed are [97], [100], and [135], whereby [97] and [135] used the same datasets as in the proposed research and [97] and [135] focused on profiling mobile devices.

In terms of the algorithm selection in the closely related works, Radhakrishnan et al. [97] selected a neural network algorithm to fingerprint devices and device types using packet inter-arrival times. However, they did not show how the data were pre-processed, and it was unclear how many training samples were used for the training and testing. Also, the choice of the algorithm was not justified. Kulin et al. [135] selected the k nearest neighbour algorithm, decision trees, logistic regression and a neural network to provide an in-depth illustration of different classification algorithms used in developing device or device-type fingerprinting. Li et al. [100] selected a neural network to profile telephony, device usage, and Bluetooth scan using publicly available data and evaluate their study by simulating device usage, telephony and Bluetooth scans to generate data. However, they did not show how the data were pre-processed and it was unclear how many training samples were used for the training and testing. Here too, the choice of the algorithm was not justified. The work presented here selected three

different algorithms, namely K-means, CMGOS and LSTM, to develop a device-type profiling technique utilising the same dataset used in [97] and [135]. First, the dataset selection, the algorithm selection, and how they were used are justified. For example, K-means clustering was used to gain insight into the dataset, and CMGOS was used in labelling the data as normal or abnormal inter-arrival time points. The LSTM was used for training and testing for the identification of abnormal inter-arrival time patterns based on device type; all of these are justified in chapters 4, 5 and 6. Lastly, the detailed experimental steps presented here can be repeated for future evaluation and comparison.

In terms of performance, the performance metrics used in [97] and [135] are accuracy, while recall and precision and equal error rates are used in [100]. The proposed work uses accuracy, recall, precision, f-score, specificity, negative predictive value and false-positive rates. The performance results reported in [97] show that the accuracy for the identification of device and device types falls between 83 and 95%, and the recall is between 54 and 94%. The performance results for the device-type identification in [135] show an accuracy between 88 and 91% and recall and precision between 46 and 99%. Meanwhile, the performance of the behaviour profiling technique in [100] achieved equal error rates between 13 and 35%. In comparison to the proposed work, the device-type IFT performed better than in the related works. The metrics (such as accuracy, recall, precision, F-score, and specificity) used in measuring the IFT outperformed, with performance accuracies (accuracy, precision, recall, etc) between 98 and 99%. Moreover, the complementary metrics used in the identification of abnormal patterns (such as NPV and FPR) outperformed the results of previous studies with very few false positives, which is normal for anomaly detection as the IFT still identifies the minimal proportion of abnormal inter-arrival time points. Also, the device-type performance is significantly better than the equal error rates reported in [100]. More importantly, the performance of the device-type IFT shows that the proposed technique can be generalised to identify abnormal device types in the cybersecurity domain, specifically for anomaly detection problems.

6.5 Chapter Summary

This chapter developed a novel intelligent filtering technique for identification of abnormal inter-arrival time patterns based on device-type. The confusion matrices presented in Figures 6.1 to 6.16 demonstrate the applicability of the intelligent filtering technique based on the numbers and percentages of correctly and incorrectly identified

abnormal inter-arrival time points for each device-type. Also, the performance of the IFT presented in section 6.3 highlights its acceptability as it was able to correctly identify most abnormal inter-arrival time points from the three different network traffic datasets based on device-type. This technique can improve network access control systems and can be adapted to overcome the NAC challenges discussed in section 2.1.4.

A particularly interesting part of the IFT is that the performances are good for all the device-types, and the testing evaluation metrics in Tables 6.3, 6.4, and 6.5 show that there is a certain similarity in their individual performances. These similarities underline that the inter-arrival time points of network traffic can be used to identify both abnormal inter-arrival time points based on device-type profiling and abnormal network behaviour in general. For example, while the network traffic measurements for the active, isolated, and passive datasets are different, the device-types available in the active and passive datasets are similar, yet the training and testing accuracy for all the device-types falls between 99.0 and 99.9%. Nevertheless, the IFT was further evaluated using precision, recall, F-score, SPC, NPV, and FPR due to the small number of abnormal inter-arrival time points for all device-types. Based on the performance results presented in section 6.3 using the aforementioned evaluation metrics, the IFT is considered successful. Although there are cases where the NPV is lower for some device-types, this is not a problem because the IFT still identifies the minimal proportion of abnormal inter-arrival time points, which is justified based on the lower FPR obtained in all the device-types. The following chapter evaluates the performance of the intelligent filtering technique by testing it in different distinct scenarios.

Chapter 7

Evaluation of Device-Type Intelligent Filtering Technique

The evaluation process of the IFT developed in this research is described here, its performance is evaluated through real and synthetic datasets. The real data are described and analysed in chapter 4, and the synthetic data were generated using empirical distribution functions, specifically the random variable histogram. The detailed synthetic data generation processes are described in section 7.1. Thereafter, the study examines the performance of the device-type IFT in two different experiments (i.e. network traffic rates and synthetic data) and present the results in sections 7.2 and 7.3. The first experiment trains the IFT using real datasets from active, isolated, and passive network traffic datasets, and a further evaluation is conducted with a synthetic dataset. Meanwhile, the second experiment involves training and evaluation based on network traffic protocol; that is, training with the UDP network traffic dataset from the isolated network traffic dataset and evaluating with the TCP datasets, and vice versa. These experiments give some insight into how well the IFT performs on all device-types for different network traffic datasets and the synthetic datasets. Finally, the chapter is summarised.

7.1 Generating Synthetic Datasets

The IFT was further evaluated using synthetic data generated from the device types previously used in the IFT implementation. As stated previously in section 4.1.1, several of the dataset repositories explored here do not have sufficient datasets that meet the requirements of this research. The criteria are that the dataset must be generated from BYOD devices (i.e. smartphones, tablets and laptops) and should

contain the important attributes outlined in section 2.3.4. However, most of the datasets investigated, such as that provided by [189], generated the data from all network devices, making it difficult to differentiate the data from BYOD devices. Given this limitation, a synthetic dataset is the only viable option to complement the real data. Moreover, the goal of generating synthetic data is to have additional data that can be used to evaluate the performance of the device-type IFT. In generating the synthetic data, an empirical and theoretical concept, specifically probability distribution fitting, was used to measure the inter-arrival time distributions and generate synthetic data based on the distributions that best fit the data for each device type. The probability distribution fitting is the fitting of a probability distribution to a series of data concerning the repeated measurement of a variable phenomenon.

The distribution fitting helps to determine the model that best fits the supplied data. The goal of fitting the distributions to a dataset is to observe the empirical data sample via a theoretical distribution model. To fit the distributions to our datasets, three different steps were followed. The first is that descriptive statistics, which give valuable indications through the histogram and skewness of the observed data, as to the choice of a suitable theoretical model. The second step uses means and standard deviations to determine the distribution parameters and to estimate the likely parameters for the empirical datasets. The third step is that of using the significance level to determine how well the observed data matches the theoretical model using the estimated parameters. The significance level is computed using a goodness-of-fit test such that if the significance level is beyond a predefined threshold, the hypothesis is accepted, or else otherwise rejected. The accepted hypothesis is when data follow the specified distribution and the rejected is when the data do not follow the specified distribution. The above three steps were applied when generating synthetic data.

7.1.1 Probability Distribution Fitting

In the synthetic data generation using probability distribution fitting, we imported the numpy, scipy, and seaborn Python libraries and data into the Pycharm IDE environment, and model the data to fit a distribution that best fits the supplied data. The first step in generating the data was fitting various distributions to the data to perform a number of checks and select the distribution that allowed for a best fit. These test distributions include normal, lognormal, cauchy, and gamma, among others [190]. After applying the above steps, we observed that the distributions did not fit our datasets and all the fitted distributions fall within different range(s). We then attempted to use the Kolmogorov Smirnov goodness-of-fit test and curve fitting to

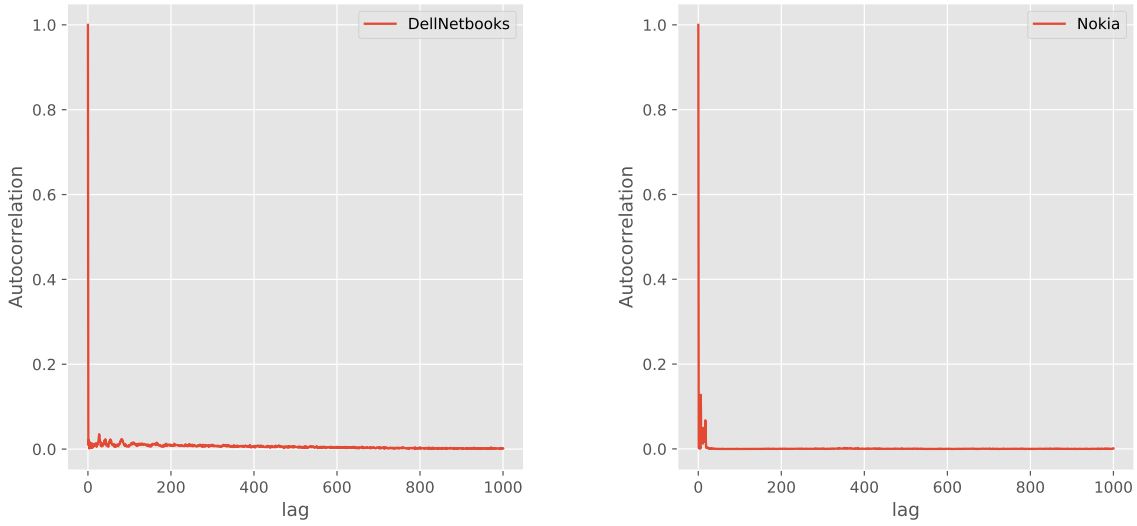
determine whether any of the data fits were acceptable [191], but the results remained essentially similar with no apparent improvement. Therefore, this was rejected and a Random Variable (RV) histogram attempted instead. The plots for the sample distributions that did not fit our datasets are presented in figures B.36 - B.38 in Appendix B.

7.1.2 Random Variable Histogram

The numpy, scipy, and seaborn Python libraries and data were imported into the Pycharm IDE environment to model and represent the data in a given histogram. The random variable histogram is a continuous variable used to generate data samples using empirical distributions. The RV histogram for the dataset containing inter-arrival times for device-type (Dev_T) defines how the IAT values are distributed over the values of a random variable(s). The following conditions were considered when generating synthetic data: it was verified that functions (IAT_x) were not taking negative values, and that the sum of the probabilities of each IAT value of the random variables lies within a certain interval available in the dataset. These conditions were met and fit the data into the distribution. The autocorrelation plots in Figure 7.1 show that there is a degree of similarity between a given inter-arrival time points and a lagged version of itself over successive time intervals, also, there is positive correlation between the real and synthetic data. Therefore, synthetic dataset can be generated from all the device-types using the RV Histogram. In this case, the synthetic dataset was generated from the device-types according to the above justification. In generating the synthetic dataset, the entire dataset for each device-type containing inter-arrival time points were used. For example, the Dell Netbooks and Nokia Phones shown in the plots had $9,100,324$ and $1,563,011$ inter-arrival time points, respectively. The $9,100,324$ inter-arrival time points for Dell Netbook were used to generate synthetic data for the Dell Netbook and the $1,563,011$ for the Nokia, with similar for the other device-types in the active, isolated, and passive network traffic datasets.

7.1.3 IFT Performance Evaluation

Before starting the evaluation, it would be useful to explain how the generated synthetic data was analysed. Here, the same algorithms and experimental settings as in sections 4.2.1 and 5.1.2 were applied to the synthetic datasets, the results of which are reported in Tables B.1 - B.17 in Appendix B. The analyses are similar to those in section 5.2. In the results, it was observed that there were no significant differences between the



(a) Dell Netbooks (b) Nokia

Fig. 7.1 Generating synthetic Data using Random Variable Histogram

analysis of the original and synthetic datasets, and therefore analysing the results will not add any value to this chapter as the main focus is to evaluate IFT using synthetic data following similar processes to those used in the original datasets.

As stated earlier, the main objective for generating a synthetic data is to evaluate the effectiveness of the IFT using the datasets that are not part of the training sets. This was conducted in two distinctive experiments, respectively. The first experiment is that of comparing real data at different speed rates (e.g. 1 Mbps and 8 Mbps) as described in section 4.1.2 and network traffic protocols (TCP, UDP, and ICMP). While, the second experiment is that of training the IFT with the real data and test using a synthetic data. These two experiments apply similar experimental settings to those in section 6.1.1.

7.2 Evaluation based on different Network Traffic Rates

The IFT was trained by taking an 80/20% split of the data for training and testing purposes, respectively, followed by additional testing with 100% of the other dataset from other network traffic measurement. For example, in Table 7.1 the active network traffic has two different datasets measured for cases 1, 2 and 3. The payload sizes for case 1 and 3 is 64 bytes and for case 2 is 1400 bytes and the speed rate for Case 1 was

measured at a speed of 1 Mbps and cases 2 and 3 at 8 Mbps, and therefore the IFT is trained with the case 1 dataset and tested with the case 2 dataset, and vice versa. A similar approach is applied for the isolated and passive network traffic datasets in case 3, but in this case apart from the speed there is an additional network protocol comparison, i.e., TCP versus UDP, ICMP versus UDP, and TCP versus ICMP, and vice versa for each.

Table 7.1 Overview of the IFT Evaluation based on different network traffic rates

Network Traffic Type	Experiment 1		Experiment 2	
	Training Dataset	Testing Dataset	Training Dataset	Testing Dataset
Active	ICMP-case 1	ICMP-case 2	ICMP-case 2	ICMP-case 1
Isolated	TCP-case 2	UDP-case 3	UDP-case 3	TCP-case 2
Passive	UDP-case 3	ICMP-case 1	ICMP-case 1	UDP-case 3

Moreover, the results of these evaluation experiments are illustrated in the receiver operating characteristic (ROC) curve in Figures 7.2, 7.3, and 7.4. The ROC curve is a metric to determine the evaluation accuracy via Area Under The Curve (AUC). It is constructed by plotting the true positive rate against the false positive rate. The x-axis show the false positive rate and the y-axis show true positive rate, and the AUC reflects the trade-offs between the FPR and TPR, where an AUC value of 0.50 indicates that the classification is equivalent to a pure random guess, and an AUC value of 1.0 indicates that the classifier perfectly distinguishes the classes (i.e., Normal and Abnormal device-type profiles). The LSTM outputs a score for each sequence (a number between 0 and 1), and if that score is $>= 0.5$, the classifier performs well without errors; if the score is < 0.5, the classifier did not perform well and gave many errors.

7.2.1 Active Network Traffic Datasets

The ROC curves for the device-types in the two active network traffic datasets are illustrated in Figure 7.2. As shown in the upper-left corner of the figure, the curves for the Asus, Acer and Gateway Netbooks, Google Phone, and Asus Tablet are relatively high, corresponding to the > 0.99 AUC value. These results are good as the IFT performed well in both experiments, although the Lenovo Laptop has a relatively low AUC (0.6614), which might be due to the two datasets (Ping-ICMP-Case 1(Lenovo) and Ping-ICMP-Case 2 (Lenovo)) being very different in nature, meaning a classifier trained on Case 1 may still be generalised on Case 2, and vice-versa. These AUC

values clearly indicate that the IFT can be generalised on all the device-types in active network traffic datasets. This also highlights that conducting the IFT training at a speed of 1 Mbps and testing at a speed of 8 Mbps, and vice versa, does not affect its performance.

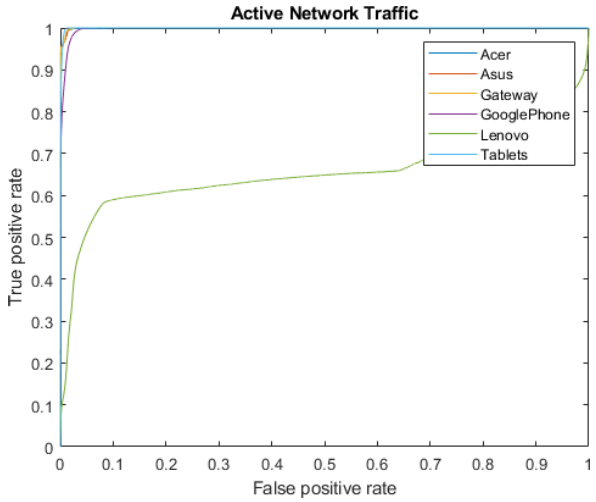


Fig. 7.2 The ROC curves for experiments 1 and 2 for the device-types in the active network traffic

7.2.2 Isolated Network Traffic Datasets

In these network traffic datasets, two experiments were conducted to determine whether the IFT could be generalised. In the first experiment, the device-types were trained on TCP case 2 and tested against UDP case 3. In the second experiment, the device-types were trained on UDP case 3 and tested against TCP case 2. As stated in section 4.1.2, the TCP case 2 device-types were measured at a speed rate of 8 Mbps while the UDP was measured at 1 Mbps. Therefore, two comparisons based on the speed rate and protocol type were conducted to assess whether the IFT can be generalised. The ROC curves for the device-types in the TCP and UDP of isolated network traffic datasets are illustrated in Figure 7.3. The device-types available in the experiments are the Nokia Phone, iPad, iPhone 3G, and iPhone 4G. As can be seen from the top-left corner of the figure, the curves for the Nokia Phone, iPad, iPhone 3G, and iPhone 4G are relatively high, whereby the iPad, iPhone 3G, and iPhone 4G were observed to have an AUC value of 0.99 while the Nokia Phone had an AUC value of 0.96. The AUC values for these device-types are good, clearly indicating that the IFT performed well and can be generalised based on these network traffic datasets due to relatively good performance across device-types.

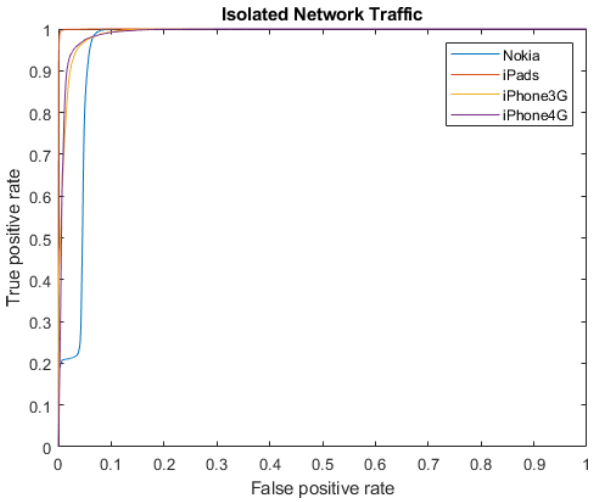


Fig. 7.3 The ROC curves for experiments 1 and 2 for the device-types in the isolated network traffic

7.2.3 Passive Network Traffic Datasets

In these network traffic datasets, we performed two experiments to explore whether the IFT can be generalised by training the IFT with the datasets measured on UDP case 3 and tested against ICMP case 1, and vice versa. The ROC curves for the device-types in the two passive network traffic datasets are illustrated in Figure 7.4.

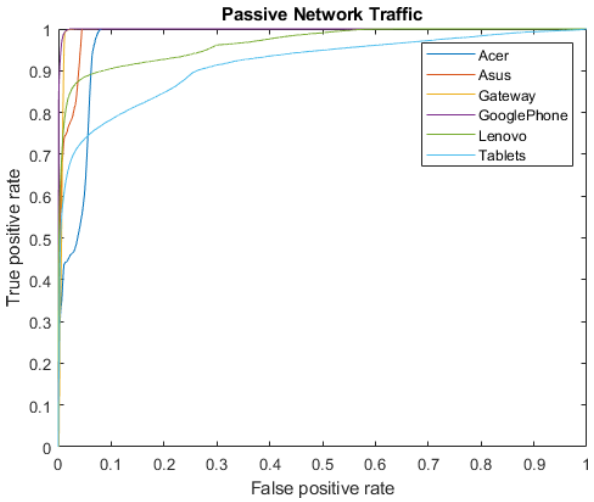


Fig. 7.4 The ROC curves for experiments 1 and 2 for the device-types in the passive network traffic

As can be seen in the upper-left corner of the figure, the curves for all device-types are relatively high, indicating a good IFT performance. The Asus and Gateway Netbooks and Google Phone had AUC values of 0.99, the Acer Netbook and Lenovo

Laptop had AUC values of 0.97, and the Asus Tablet had the lowest AUC value of 0.92. The results show that the IFT performed well for all device-types and can be generalised on the device-types in these network traffic datasets.

7.3 Evaluation based on Synthetic Datasets

Similar experimental settings to those in section 6.1.1 with an additional testing step with 100% synthetic datasets was applied to evaluate the effectiveness of using a data that was not part of the training set. The overview of the training and testing for each device-type used for IFT evaluation using synthetic dataset is illustrated in Table 7.2.

Table 7.2 Overview of the IFT Evaluation based on synthetic datasets for the Device-Types in Active Network traffic Datasets.

		Real Datasets		Synthetic Datasets
Network Traffic Type	Device Type	Training IATs (80%)	Testing IATs (20%)	Testing IATs (100%)
Active	Acer NB	3,174,825	793,718	3,968,543
	Asus NB	3,174,850	793,975	3,968,825
	Gateway NB	2,543,935	635,996	3,179,931
	Google Phone	637,405	159,363	796,768
	Lenovo Laptop	637,613	159,662	797,275
	Asus Tablet	635,931	158,995	794,926
Isolated	iPad	3,655,182	916,308	4,571,490
	iPhone 3G	903,479	225,870	1,129,349
	iPhone 4G	6,640,562	1,660,153	8,300,715
	Nokia	1,250,360	312,602	1,562,962
Passive	Acer NB	2,569,901	642,487	3,212,388
	Asus NB	2,569,713	642,440	3,212,153
	Gateway NB	2,055,754	513,951	2,569,705
	Google Phone	520,123	130,043	650,166
	Lenovo Laptop	514,021	128,517	642,538
	Asus Tablet	513,953	128,500	642,453

The most important aspect of evaluating the effectiveness of the IFT is its ability to predict the correct output classes. This was measured using the accuracy, recall, precision, and F-score, SPC, NPV, and FPR evaluation metrics with their metrics equations defined in section 6.3, which illustrates the performance of the IFT identification on the synthetic datasets. This metrics provides quantifiable evidence of how

effective the IFT is at making correct predictions. The IFT evaluation results for all the network traffic datasets experimented, using the considered evaluation metrics for the device-types, are presented in Tables 7.3, 7.4, and 7.5. The tables show the normal and abnormal inter-arrival time points used for testing the IFT with synthetic data and the evaluation metrics for each device-types. In the tables, we observe the IFT to have outperformed in all the device-types with evaluation accuracy, recall, precision, and F-Score between 97.0 to 99.9%.

7.3.1 Active Network Traffic Datasets

The IFT evaluation results for the active network traffic datasets, based on the considered evaluation metrics for the studied device-types, are presented in Table 7.3. The device-types analysed are the Acer, Asus, and Gateway Netbooks, the Lenovo Laptop, and the Asus Tablet in the ICMP case 1 dataset, enabling the results to be compared with the other network traffic datasets. Meanwhile, the results for the other device-types are similar to what was found for the other device-types in the ICMP case 2 datasets; therefore, they are also part of the analysis and comparison. The overall accuracy, precision, recall, and F-score are similar for all device-types, which is a good sign that the IFT identification outperforms when testing with synthetic data. In addition, to ensure the correctness of the IFT identification, other complementary metric equations, such as SPC, NPV, and FPR, as described in section 6.3, are used to measure the IFT performance for the device-types in the active network traffic datasets.

Table 7.3 Testing Evaluation Metrics with Synthetic Data for the Device-Types in the Active Network

	Inter-arrival Time (s)								
Device	Normal (%)	Abnormal (%)	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	SPC (%)	NPV (%)	FPR (%)
Acer NB	3,967,663	880	99.7	99.9	99.8	99.8	92.7	98.2	7
Asus NB	3,967,892	933	99.7	99.9	99.8	99.8	91.6	79.8	8
Gateway NB	3,177,727	2,204	98.1	99.9	98.1	99.8	99.1	3.4	0.9
Google Phone	775,750	21,018	99.1	99.8	99.3	99.5	92.2	78.2	8
Lenovo Laptop	776,498	20,777	99.5	99.7	99.8	99.7	89.3	90.9	11
Asus Tablet	775,869	19,057	99.6	99.9	99.6	99.4	84.5	72.9	13

Table 7.3 shows the synthetic data testing evaluation results for the device-types in the active network traffic datasets. The table shows that the IFT outperforms with

the synthetic data in all three metrics for all device-types, with the IFT identification accuracy surpassing 98.1%. Among all the device-types, the Acer and Asus Netbooks obtain the highest values for accuracy, recall, precision, and F-score, which means that the IFT correctly identified the inter-arrival time points in both the normal and abnormal device-type profiles while incorrectly identifying between 0.1 and 0.3% of the inter-arrival points in both profiles. Meanwhile, the Asus Tablet has the second-highest accuracy, at 99.6%, which indicates that the IFT correctly identified 99.6% of the inter-arrival points, with 0.4% being incorrectly identified. Besides, the Asus Tablet has 99.9% precision and 99.6% recall, indicating that the IFT identified not only the normal profile but also the abnormal profile. The performance of the other device-types falls between 98.1 and 99.5%, meaning that the IFT correctly identified between 98.1 and 99.5% of the inter-arrival time points while incorrectly identifying between 0.5 and 1.9%. Moreover, the precision, recall, and F-score values show that the IFT correctly identified 99.3 and 99.8% of the inter-arrival time points in the normal and abnormal profiles, respectively.

In terms of the false positive alarm rates, the IFT is further evaluated using SPC, NPV, and FPR. As can be seen from the table, the SPC for all the device-types falls between 84.5 and 99.1%, whereby the Google Phone has the highest SPC at 99.2% and the Asus Tablet has the lowest SPC at 84.5%. Meanwhile, the NPV for the device-types ranges between 72.9 and 98.2%, except for the Gateway Network, which has the lowest NPV (3.4%). Within this range (72.9 to 98.2%), the Acer Netbook has the highest NPV and the Asus Tablet has the lowest NPV. Judging by the NPV values, we can conclude that the IFT identification outperforms for the rest of the device-types, except for the Gateway Netbook, which requires further investigation. Furthermore, the FPR ranges between 0.3 and 2%, with the Gateway Netbook having the lowest (0.9%) and the Asus Tablet having the highest (13%) values. The FPR is not high, which means that the IFT is not biased towards false positive or false negative classes but rather correctly identifies both normal and abnormal inter-arrival time points. Hence, the FPR rules out the NPV for the Gateway Netbook and the other device-types that have lower NPVs. Based on the evaluation metrics, it is concluded that the IFT outperforms for all the device-types in the active network traffic datasets and can thus be generalised, with similar results being observed for the device-types in the passive ICMP case 2 datasets.

7.3.2 Isolated Network Traffic Datasets

The IFT evaluation results for the Isolated Network Traffic Datasets, based on the considered evaluation metrics for the available device-types, are presented in Table 7.4. The iPad, iPhone 3G, iPhone 4G, and Nokia in the TCP case 2 dataset are analysed to enable a comparison of the results. Furthermore, as the results for the other device-types in the UDP case 3 dataset are similar to those found for the device-types examined, they are also part of the analysis and comparison with the other device-types in the active and passive network traffic datasets. The overall accuracy, recall, precision, and F-score are similar for all the device-types, whereby the best performance was observed for the iPhone 3G and iPhone 4G, followed by the iPad and Nokia Phone. Also, the SPC, NPV, and FPR described in section 6.3 are used to measure the IFT performance for the device-types in the isolated network traffic datasets.

Table 7.4 Testing Evaluation Metrics with Synthetic Data for the Device-Types in the Active Network

	Inter-arrival Time (s)								
Device	Normal (%)	Abnormal (%)	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	SPC (%)	NPV (%)	FPR (%)
iPads	4,562,847	8,643	99.7	99.9	99.8	99.8	53.8	99.9	46
iPhone 3G	1,113,955	15,394	97.1	98.5	97.1	97.8	98.1	31.6	2
iPhone 4G	8,258,186	42,529	99.5	99.8	99.5	99.6	99.2	51.8	0.8
Nokia	1,562,486	476	99.7	99.9	99.8	99.8	33.6	99.9	66

Table 7.4 presents the synthetic data testing evaluation results for the iPad, iPhone 3G, iPhone 4G and Nokia Phone. According to the table, the IFT correctly identified 99.7% of the inter-arrival time points for the iPad, while incorrectly identifying 0.3%. In terms of precision and recall, the IFT identified 99.9% of the inter-arrival time points in both the normal and abnormal profiles. For the iPhone 3G, the accuracy is 97.1%, meaning that the IFT correctly identified 97.1% of the inter-arrival time points while incorrectly identifying 2.9%. Here, the recall is similar to the accuracy and the precision is 98.5%, which is an indication that the IFT correctly identified the inter-arrival time points as categorised into normal and abnormal profiles in both the actual and predicted outputs. Moreover, the performance for the Nokia Phone is similar to that for the Acer and Asus Netbooks, whereas for the iPhone 4G it is similar to that for the Lenovo Laptop in the active network traffic datasets.

Regarding the false positive alarm rates, the IFT is further measured using SPC, NPV, and FPR. The table shows that the SPC for all the device-types falls between 33.6 and 99.2%, whereby the Nokia Phone has the lowest SPC at 33.6% and the iPhone 4G has the highest SPC at 99.2%. Meanwhile, the NPV for the device-types is 31.6 and 51.8% for the iPhone 3G and iPhone 4G, respectively, and is 99.9% for both the iPad and the Nokia Phone. Based on the NPV, it is clear that the IFT identification outperforms for the iPad and Nokia Phone but does not perform well for the iPhone 3G and iPhone 4G. Beyond that, the FPR falls into two ranges (i.e. 0.8 to 2% and 46 to 66%), whereby the lower range shows the percentage of the device-types for which the IFT outperforms, while it does not perform well for those in the higher range. In this case, the iPhone 3G and iPhone 4G have the lowest FPR and the Nokia Phone and iPad have the highest, which is the opposite of the NPV results. Hence, the NPVs for the iPhone 3G and iPhone 4G are ruled out by the FPR. The FPR is not high in the case of the iPhone 3G and iPhone 4G, which means that the IFT is not biased towards false positive or false negative classes but rather correctly identified both normal and abnormal inter-arrival time points. Moreover, it does not perform well for the iPad and Nokia Phone due to the lower SPC and higher FPR. Indeed, based on the evaluation metrics used, it is concluded that the IFT outperforms for the iPhone 3G and iPhone 4G in the isolated network traffic datasets and can be generalised, while further investigation is required for those device-types for which it produced large false alarms (iPad and Nokia Phone). Similarly, the testing results for the device-types in the isolated UDP case 3 dataset are not different from the results analysed here or for the active network traffic datasets.

7.3.3 Passive Network Traffic Datasets

Based on the considered evaluation metrics for the available device-types, the IFT evaluation results for the passive network traffic datasets are presented in Table 7.5. This section analyses the Acer, Asus, and Gateway Netbooks, the Lenovo Laptop, and the Asus Tablet in the UDP case 3 dataset to facilitate a comparison of the results with the device-types in the ICMP case 1 dataset as well as the active and isolated network traffic datasets. The overall accuracy, recall, precision, and F-score are similar for all device-types, indicating the fact that the IFT identification outperforms when testing with passive network traffic synthetic data. Furthermore, to verify the correctness of the IFT identification, complementary metric equations, namely SPC, NPV, and FPR, as described in section 6.3, are used to evaluate the IFT performance for the device-types in the active network traffic datasets.

Table 7.5 Testing Evaluation Metrics with Synthetic Data for the Device-Types in the Active Network

	Inter-arrival Time (s)								
Device	Normal (%)	Abnormal (%)	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	SPC (%)	NPV (%)	FPR (%)
Acer NB	3,193,659	18,729	99.8	99.9	99.8	99.8	97.2	79.1	3
Asus NB	3,202,827	9,326	99.8	99.9	99.8	99.8	97.9	59.7	2
Gateway NB	2,556,156	13,549	99.8	99.9	99.8	99.8	98.4	77.2	2
Google Phone	636,873	13,293	97.0	99.8	97.1	98.4	91.4	39.4	9
Lenovo Laptop	635,962	6,576	98.0	99.5	98.5	98.9	47.7	25.0	52
Asus Tablet	625,987	16,466	99.5	99.9	99.6	99.7	95.1	87.0	5

Table 7.5 shows the synthetic data testing evaluation results for the device-types in the passive network traffic datasets. The table demonstrates that the IFT outperforms with the synthetic data in all the three metrics for all the device-types, whereby the IFT identification accuracy surpasses 97%. Of the device-types, the Acer, Asus and Gateway Netbooks obtain the highest values in terms of accuracy, recall, precision and F-score, showing that the IFT correctly identified 99.8% of the normal and abnormal inter-arrival time points for these device-type profiles, with between 0.2% inter-arrival points being incorrectly identified. The performance for the Asus Tablet was slightly lower, with an accuracy of 99.5%, which indicates that the IFT correctly identified 99.5% of the inter-arrival points and incorrectly identified 0.5%. Furthermore, the 99.9% precision, 99.6% recall, and 99.8% F-score indicate that the IFT identified not only the normal profile but also the abnormal profile. The performance for the Lenovo Laptop and Google Phone falls between 97 and 98%, highlighting that the IFT correctly identified between 97 and 98% of the inter-arrival time points while incorrectly identifying between 2 and 3%. Moreover, the precision, recall and F-score show that the IFT correctly identified between 97.1 and 99.5% of the inter-arrival time points in the normal and abnormal profiles, respectively.

In terms of the false positive alarm rates, the IFT is further evaluated using SPC, NPV, and FPR. As per the table, the SPCs for all the device-types fall between 47.7% and 98.4%, whereby the Lenovo Laptop has the smallest SPC (47.7%) and the Gateway Netbook has the highest SPC (98.4%). Meanwhile, the remaining device-types fall within the range of 95.1 to 98.4%. As for the NPV, it ranges between 25.0 and 87.0%, with the Lenovo Laptop and Google Phone having 25% and 39%, respectively, which are the lowest values among all the device-types, followed by the Asus Netbook with 59.7%; the remaining device-types have between 77.2 and 87.0%. Based on the NPV,

the IFT identification outperforms for the rest of the device-types, although in the case of the Lenovo Laptop and Google Phone, further investigation is needed. In addition, the FPR ranges between 3 and 9%, except for the Lenovo Laptop, which has the highest FPR (53%) among the device-types evaluated here. An FPR between 3 and 9% is not high, which means that the IFT is not biased towards false positive or false negative classes but instead correctly identified both normal and abnormal inter-arrival time points. Hence, the FPR rules out the NPV for the Google Phone and the other device-types with a lower NPV. However, the Lenovo Laptop could not be ruled out due to the lower SPC and NPV observed for this device-type. Finally, based on the evaluation metrics used, it is concluded that the IFT outperforms for the majority of the device-types examined in the active network traffic datasets and can be generalised, with similar results being observed for the device-types in the passive ICMP case 1 datasets.

7.4 Chapter Summary

The first part of this chapter discussed the generation of synthetic data using different synthetic data-generating techniques. The random variable histogram was used to generate synthetic datasets because this was found to give the best fits to our datasets. Algorithms such as K-means clustering and clustering-based multivariate gaussian outlier score were applied to the generated synthetic data to prepare and obtain a labelled dataset that was used to evaluate the intelligent filtering technique. The second part of the chapter evaluates the IFT based on different network traffic rates, whilst the third part analysed the evaluation results based on synthetic data. The results of the evaluation using the well-known classification metrics used for the intelligent filtering technique, as presented above, demonstrate the accuracy, robustness, and effectiveness of intelligent filtering technique in correctly identifying the abnormal inter-arrival time points for each device-type.

The evaluation results for the original datasets (first experiment) based on speed and network protocol comparison, as analysed in section 7.2, clearly show that the IFT performed well for all device-types although Lenovo Laptop had lower AUCs of 0.66 and 0.92, which is not a problem as the AUC values are greater than 0.5 . The AUCs for other device-types ranged between 0.97 to 0.99. The analysis of the IFT with synthetic data (second experiment), as presented in section 7.3, showed that the IFT correctly identified the abnormal inter-arrival time points for all device-types with an accuracy between 90 to 99%. From the analysis, it was clear that some device-types have a

lower NPV, which is not a problem in anomaly detection as the lower FPR normally ruled that out especially in cases where the datasets are not balanced. Additionally, the results for both experiments did not suggest that any one of the evaluations was better than the others, but did show the effectiveness and robustness of the IFT and, indeed, that it can be generalised.

Chapter 8

Conclusion and Future Work

An IFT is developed in this thesis as a technique for the identification of abnormal network traffic pattern(s) based on the device-type. In the IFT implementation, three algorithms, namely K-means clustering, CMGOS, and long short-term memory, as described and justified in chapter 3, are used. This chapter outlines how the research objectives are met. First, section 8.1 introduces the overall problem(s) addressed in this work. Then, the contributions made in this research are presented in section 8.2, which also presents further work that could be carried out to advance what has been achieved and the limitations of the research are highlighted in section 8.3 and section 8.4 presents the future work. Finally, the chapter is summarised in section 8.5.

8.1 Introduction

Network Access Control (NAC) is an underlying system for cybersecurity defence. The premise of NAC is to provide security for end-to-end solutions based on policy-based access to different parts of the enterprise network by blocking or quarantining the devices that do not comply with a set of predefined security policies or provide an indication of vulnerabilities in the network. Also, NAC solutions provides endpoint visibility after device data has passed into the network defence system but before the data is stored on a storage system. NAC has gained wider acknowledgement and use following the rise of Bring Your Own Device (BYOD) trend, so that enterprise networks no longer depend solely on the traditional security measures [192]. This is because the traditional security measures do not sufficiently protect enterprise networks from cyber-attacks [193]. NAC security standards operate in IEEE 802.1x protocols to define and encapsulate network traffic based on the Extensible Authentication Protocol (EAP) over IEEE 802. These are established standards based on traffic encryption and

integrity that protects network infrastructure as well as preventing unauthorised and illegal access to enterprise network. The 802.1x security cannot rely purely on security protocols to protect network traffic during data communication. Therefore, achieving a robustly secure and reliable BYOD network design poses a huge challenge to enterprise networks. It has become highly crucial to develop and adopt an improved approach to strengthen the security level in evolving enterprise network systems.

This research has considered a crucial security issue inherent in NAC that relates to the limitation or failure to recognise and/or prevent the use of genuine devices by unauthorised users to access or compromise the enterprise networks. As a solution, this research proposes an intelligent filtering technique (IFT) that addresses this concern by looking into the variation of packet inter-arrival time patterns of the device-types connected to the enterprise networks. This can be used to identify and filter abnormal network traffic patterns or devices that suggest malicious activity and to help control access. Different kinds of security issues can cause abnormal network traffic in NAC systems that can compromise enterprise networks. Initial literature study reveals that different kinds of security issues can cause abnormal network traffic in NAC systems that can compromise enterprise networks. Some of these issues include; identity theft and unauthorised access. Initial literature study also presents some of the key security requirements for this kind of system, such as ensuring confidentiality, integrity, and availability. The literature study also explored the possibility of finding a common pattern that could be used to establish preventive measures against the security issues identified which can cause abnormal network traffic. The most commonly used countermeasures identified in the literature reviewed are behaviour profiling and fingerprinting techniques [2] which vary, but both rely on identification or profiling abnormal patterns. However, neither behavioural profiling nor fingerprinting in isolation provides a sufficiently robust preventive measure against device network access control threats. While fingerprinting techniques focus on reconnaissance, i.e., gathering information about devices accessing the network, current behavioural profiling techniques typically focus on device users. This research combines the two countermeasure techniques to achieve a more robust approach to protect enterprise networks from intrusion. This is achieved by identifying a device-type and profiling the device to identify abnormal network traffic patterns that suggest malicious activity from packet inter-arrival times, and use same to control access.

8.2 Contributions

The main contribution of this research is the development of a novel security technique for Bring Your Own Device-based network access control (NAC) systems. The security technique is unique in that it employs device-type profiling technique to distinguish abnormal (malicious) from normal (benign) network traffic. No prior work has been found to adopt this approach for NAC systems. In the novel approach proposed, a unique inter-arrival time data analysis using K-means and notched box plots was developed. A device-type profiling technique was developed using the clustering-based multivariate gaussian outlier scores. The identification of abnormal network traffic detection was achieved using long short-term memory networks. The device type intelligent filtering assumption and the way it is used in defining these algorithms are also novel. The overview of the main contributions of this research and how the research objectives and questions were answered is presented in Figure 8.1. The figure shows the series of stages taken, from the literature review to the evaluation of the device-type IFT described in detail below.

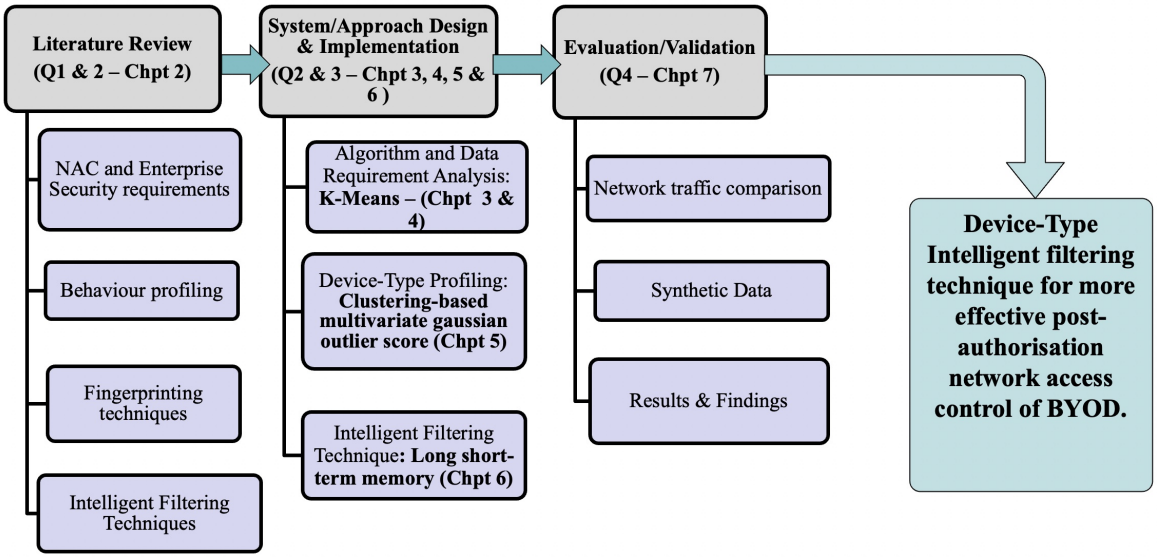


Fig. 8.1 An overview of the research methodology

1. Literature review: In chapter 2, An overview of the inherent security threats, vulnerabilities, attacks and security requirements associated with BYOD-enabling technologies is presented. Current measures in the domain for addressing security risks and requirements are also described. The chapter starts by introducing the NAC and enterprise security requirements. Then, it reviews the related works in behaviour profiling, fingerprinting techniques and IFTs to determine

the weaknesses in the related works as well as address the identified weaknesses. In relation to the research objectives and questions, the literature chapter review addresses research objectives 1 and 2 as well as research questions 1 and 2.

2. Description and justifications of algorithms: chapter 3 investigated the appropriate machine learning techniques for outlier detection along with the justification for selecting the technique used throughout the thesis. In relation to the research objectives and questions, the methodology justification chapters addressed research objectives 1 and 2, as well as research questions 1 and 2. The clear contributions to knowledge of this work are covered in chapters 4 to 7 of this thesis report, and are summarised as follows:

3. Data analysis: In chapter 4, The approach used to analyse the data and the experimental settings are both unique and present an improvement to existing concepts available in this domain. Current works use clustering techniques to assume normal and abnormal patterns. Other researchers use classification techniques, e.g. neural networks, to classify normal and abnormal patterns. The new approach proposed in this study uses K-means clustering to understand the inter-arrival time patterns of the data from the devices before defining or classifying normal and abnormal patterns. No known technique has been found in this area that uses this type of data analysis approach (using K-means to understand the inter-arrival time patterns, and notched box plots to validate the device-type profiling approach) to assess datasets containing inter-arrival time data values for multiple mobile devices from three different network monitors (active, isolated and passive). In relation to the research objectives and research questions, the data analysis chapter addressed research objective 3 and a part of research question 3.

4. Device type-profiling using packet inter-arrival times: This is one of the contributions of this research and is presented in chapter 5. It includes the definition of the algorithm and a device-type profiling technique developed to classify and label the device type datasets to distinguish abnormal from normal inter-arrival times. The main idea behind this approach is to use the clusters identified in the clustering algorithm to profile, classify, and label the normal and abnormal inter-arrival time points for each device type. In doing so, we use the output of the clustering algorithm by feeding this into the input of a clustering-based multivariate gaussian outlier score and then calculate how likely a data instance is to be close to the cluster centre. For example, the data values associated with,

or close to, each cluster are added to where they belong. The outlier score for each data instance is later computed based on the multivariate gaussian of each cluster, an outlier score is added to inter-arrival time points, and each point is labelled either normal or abnormal following the standard defined by Goldstein et al. [143]. This device type profiling will enable system administrators to clearly group the packet inter-arrival times into normal and abnormal classes for labelling and model training. In relation to the research objectives and research questions, the device-type profiling chapter addresses research objective 4 and answered a part of research question 3.

5. Intelligent filtering technique based on device type: This novel contribution covered in chapter 6, uses the bidirectional architecture of long short-term memory network due to its dynamic filtering capability in training the IFT. The idea behind this approach is that the labelled dataset containing the inter-arrival time values and target (normal or abnormal) inter-arrival time points from the device types is used to train the IFT. The IFT has been shown to correctly identify the abnormal inter-arrival time points from all the device types with an accuracy above 99%. In relation to the research objectives and research questions, the device type profiling chapter successfully answered research question 4 and addressed research objective 5. Also, the evaluation results in chapter 7 showed the effectiveness of the IFT for different forms of network traffic and protocols. The evaluation demonstrates that the device type IFT can be generalised for similar datasets, as demonstrated and justified using the two different scenarios in the evaluation chapter.

This research and the novel outputs have made positive contributions to the BYOD enterprise network domain evidenced by their acceptance and publication at conference and in peer reviewed journals. One of the conference papers was selected based on the reviewers' comments and invited to be extended for submission to a special issue of the Journal of Sensors and Actuators. Although it was not extended at the time, it is currently being revised with the intention to submit to another journal.

8.3 Limitations of the research

The network access control for BYOD enterprise network domains has been improved. However, despite the objectives of this thesis having been met, there are several limitations associated with this research that were beyond the author's control. Some

of these limitations were mostly related to the datasets, which were not labelled with normal and abnormal inter-arrival time points. Therefore, it was difficult to differentiate the abnormal from the normal inter-arrival time points. This was addressed in this research by applying K-means clustering to identify the normal and abnormal inter-arrival time points and using CMGOS to label and classify the normal and abnormal inter-arrival time points. However, there is still a need for a labelled dataset that has normal and abnormal inter-arrival time points to ensure the correct identification of abnormal patterns. To address this limitation, an experiment needs to be conducted on real BYOD devices, such as smartphones, tablets and laptops, to generate normal data. In addition, an attack needs to be conducted on the aforementioned BYOD devices to generate abnormal data. Performing such an experiment on real devices has a cost implication in terms of buying the mobile devices and computational complexity, which is also a limitation. To address the cost implication, a limitation testbed can be implemented to generate normal and abnormal data, although testbeds may have performance issues, such as slow data generation, which would have implications in anomaly detection as the normal inter-arrival times could behave abnormally. The closest solution to address the cost implication is to configure virtual machines using the images of smartphones, tablets and laptops based on their model and operating systems, which could be reflected in future work.

8.4 Future Work

This research has improved the network access control for BYOD enterprise network domains. However, there are some areas where further work could be carried out to advance upon what has been achieved in this research. Although the objectives of this thesis having been met, the outcome of this research can be further improved with the availability of higher computational resources (i.e., random access memories' capability to handle large data) to apply the techniques for all the datasets described in section 4.1.2. More specifically, other areas of further work related to this research can be the development of an IFT into a mobile application that can intelligently detect abnormal patterns based on the device-type in a BYOD enterprise network environment.

Furthermore, the device-type IFT needs the attention of a network administrator to monitor and respond to abnormal inter-arrival time patterns. Hence, there is need to develop an alert system based on inter-arrival times that can inform the network administrator about intrusions. Such an alert system would need to have components that can automatically pre-process the abnormal inter-arrival time patterns learned

by the IFT and cluster, and then merge the alerts. Further research could explore and develop a technique to block the abnormal inter-arrival time patterns identified by the IFT. Another research direction involves investigating the application and efficiency of the device-type IFT in a real BYOD enterprise network. The application can be developed as an add-on security module that can be added to NAC systems or configured on hardware to function as an intelligent filter. This can help to consolidate the validity and reliability of device-type IFT to detect and prevent intrusions in a real network environment. It could also help to discover new parameters that may not have been considered in the development of the IFT.

8.5 Chapter Summary

The overall objectives of this research as originally specified in the first chapter have been met. The research questions are answered in the form of chapters, which describe a series of experimental studies and synthetic data generation undertaken for the profiling of mobile devices according to device type and developing an IFT to identify abnormal network traffic pattern(s) based on device type. The main contribution of this thesis is a new comprehensive security technique for the identification of abnormal network traffic pattern(s) in BYOD enterprise networks. This security technique is unique as no similar approach exists. The results produced here for the IFT identification training and testing underscore its potential in identifying abnormal network traffic patterns in BYOD enterprise networks. Moreover, the evaluation results show that the IFT can be generalised to provide a more robust underlying security for BYOD enterprise networks and can also be expanded to function as a hardware device. BYOD security concerns often arise due to a lack of sufficiently innovative methods to control employees' devices accessing enterprise networks, thereby rendering these mobile devices vulnerable to malicious attackers. For example, if an attacker can gain possession of a lost or stolen device, they become, in effect, an internal user/employee, allowing them to access the sensitive information contained on the device and cause considerable damage to the BYOD enterprise network. BYOD is a new paradigm that requires advanced security countermeasures, and the device-type IFT developed here offers a new security-counter measure that will enhance BYOD network access control, ultimately reducing the impact of unauthorised and illegal access as well as insider threats to BYOD enterprise networks. With the implemented technique, BYOD enterprise networks will have enhanced security that protects not only users/employees but also the devices connected to the networks. Moreover, this solution can also be implemented to address

anomaly detection problems in other security-related areas, such as industrial control systems, the Internet of Things, and health care systems, among others.

The research has led to four publications, and further articles will be published later from this thesis. Finally, the chapter introduced the problems addressed in thesis in section 8.1, the research contributions made in section 8.2, the future research directions in section 8.3 as well as summary of the chapter.

References

[1] Musa Abubakar Muhammad, Pooneh Bagheri Zadeh, and Aladdin Ayesh. Improving security in bring your own device (byod) environment by controlling access. In *Faculty of Technology Conference*, pages 1–4, Leicester, UK, 2017. DMU, Dora.

[2] Musa Abubakar Muhammad, Aladdin Ayesh, and Pooneh Bagheri Zadeh. Developing an intelligent filtering technique for bring your own device network access control. In *Proceedings of the first International Conference on Future Networks and Distributed Systems*, ICFNDS '17, pages 1–8, Cambridge, UK, 2017. ACM.

[3] Musa Abubakar Muhammad and Aladdin Ayesh. A behaviour profiling based technique for network access control systems. *International Journal of Cyber-Security and Digital Forensics*, 8(1):23–30, 2019.

[4] Musa Abubakar Muhammad, Aladdin Ayesh, and Isabel Wagner. Behavior-Based Outlier Detection for Network Access Control Systems. In *Proceedings of the 3rd International Conference on Future Networks and Distributed Systems*, ICFNDS '19, pages 1–6, Paris, France, 2019. ACM. ISBN 978-1-4503-7163-6. doi: 10.1145/3341325.3342004.

[5] Mohammed Ketel and Thomas Shumate. Bring your own device: Security technologies. In *Proceedings of the IEEE SoutheastCon 2015, April 9 - 12, 2015 - Fort Lauderdale, Florida*, pages 1–7. IEEE, 2015.

[6] U. Vignesh and S. Asha. Modifying security policies towards byod. *Procedia Computer Science*, 50(1877-0509):511–516, 2015. doi: https://doi.org/10.1016/j.procs.2015.04.023. URL https://www.sciencedirect.com/science/article/pii/S1877050915005244. Big Data, Cloud and Computing Challenges.

[7] Nungki Selviandro, Gede Wisudiawan, Shinta Puspitasari, and Monterico Adrian. Preliminary study for determining bring your own device implementation framework based on organizational culture analysis enhanced by cloud management control. *2015 3rd International Conference on Information and Communication Technology (ICoICT)*, pages 113–118, 2015. doi: 10.1109/ICoICT.2015.7231407. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7231407.

[8] Rob Smith, Bryan Taylor, Chris Silva, Manjunath Bhat, Terrence Cosgrove, and John Girard. Magic quadrant for enterprise mobility management suites, 2016. URL https://www.gartner.com/doc/reprints?id=1-390IMNG&ct=160608&st=sb.

[9] Jack Madden and Brian Madden. *Enterprise Mobility Management: Everything you need to know about MDM, MAM, and BYOD*. Jack Madden, 2013. ISBN 9780989650601. URL https://books.google.co.uk/books?id=wXsMngEACAAJ.

[10] Yvette E Gelogo and Haeng-Kon Kim. Mobile integrated enterprise resource planning system architecture. *International Journal of Control and Automation*, 7(3):379–388, 2014.

[11] Mark R Waterfill and CA Dilworth. Byod: Where the employee and the enterprise intersect. *Employee Relations Law Journal*, 40(2):26–36, 2014.

[12] Kathleen Downer and Maumita Bhattacharya. Byod security: A new business challenge. In *2015 IEEE International Conference on Smart City/SocialCom/-SustainCom (SmartCity)*, pages 1128–1133. IEEE, 2015.

[13] A. Bhat, V. Gojanur, and R. Hegde. 4G protocol and architecture for byod over cloud computing. In *International Conference on Communications and Signal Processing (ICCSP)*, pages 0308–0313, April 2015. doi: 10.1109/ICCSP.2015.7322894.

[14] M. Eslahi, M. V. Naseri, H. Hashim, N. M. Tahir, and E. H. M. Saad. Byod: Current state and security challenges. In *Computer Applications and Industrial Electronics (ISCAIE), 2014 IEEE Symposium on*, pages 189–192, April 2014. doi: 10.1109/ISCAIE.2014.7010235.

[15] N. Leavitt. Today's mobile security requires a new approach. *Computer*, 46(11): 16–19, November 2013. ISSN 0018-9162. doi: 10.1109/MC.2013.400.

[16] Gangadharan Byju Pularikkal, Santosh Ramrao Patil, Mark Grayson, and Madhusudan Nanjanagud. Mobile communications over secure enterprise networks, January 31 2019. US Patent App. 15/854,181.

[17] Steven M Willens. Network access control system and process, March 30 1999. US Patent 5,889,958.

[18] Antonio Scarfo. New security perspectives around byod. In *Broadband, Wireless Computing, Communication and Applications (BWCCA), 2012 Seventh International Conference on*, pages 446–451. IEEE, 2012.

[19] Abhinav BANSAL and Purvi Desai. Multidimensional risk profiling for network access control of mobile devices through a cloud based security system, March 5 2019. US Patent App. 10/225,740.

[20] Sanjay Kumar Rai, Pankaj Mishra, Shivaji Kumar Yadav, and Manik Chandra Pandey. *Cyber Security*. Book Bazooka Publication, 2019.

[21] Simon John Haswell. Network access control, May 23 2019. US Patent App. 16/096,546.

[22] Thomas Shumate and Mohammed Ketel. Bring your own device: Benefits, risks and control techniques. In *Proceedings of the IEEE SoutheastCon 2014*, pages 1–6. IEEE, 2014.

[23] Lau Lap Bann, Manmeet Mahinderjit Singh, and Azman Samsudin. Trusted Security Policies for Tackling Advanced Persistent Threat via Spear Phishing in BYOD Environment. *Procedia Computer Science*, 72:129–136, 2015. ISSN 18770509. doi: 10.1016/j.procs.2015.12.113. URL http://linkinghub.elsevier.com/retrieve/pii/S1877050915035747.

[24] P C Castro, J W Ligman, M Pistoia, J Ponzo, G S Thomas, S P Wood, and M Baluda. Enabling Bring-Your-Own-Device using mobile application instrumentation. *IBM Journal of Research and Development*, 57(6):1–11, 2013. ISSN 00188646. doi: 10.1147/JRD.2013.2279640.

[25] J Morris Chang, Pao-Chung Ho, and Teng-Chang Chang. Securing byod. *IT Professional*, 16(5):9–11, 2014.

[26] Morufu Olalere, Mohd Taufik Abdullah, Ramlan Mahmod, and Azizol Abdullah. A review of bring your own device on security issues. *SAGE Open*, 5(2): 2158244015580372, 2015.

[27] Phillip Dawson. Five ways to hack and cheat with bring-your-own-device electronic examinations. *British Journal of Educational Technology*, 47(4):592–600, 2016.

[28] Khoula AlHarthy and Wael Shawkat. Implement network security control solutions in BYOD environment. *Proceedings - 2013 IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2013*, pages 7–11, 2013. doi: 10.1109/ICCSCE.2013.6719923.

[29] Dongwan Kang, Joohyung Oh, and Chaetae Im. A study on abnormal behavior detection in byod environment. *World Academy of Science, Engineering and Technology, International Journal of Environmental, Chemical, Ecological, Geological and Geophysical Engineering*, 7(12):893–896, 2013.

[30] Aparna Bhat, Vishwanath Gojanur, and Rajeshwari Hegde. 4g protocol and architecture for byod over cloud computing. In *International Conference on Communications and Signal Processing (ICCSP)*, pages 0308–0313. IEEE, 2015.

[31] Bo Sun, Fei Yu, Kui Wu, and Victor Leung. Mobility-based anomaly detection in cellular mobile networks. In *Proceedings of the 3rd ACM workshop on Wireless security*, pages 61–69. ACM, 2004.

[32] Juniper Networks. Trusted Mobility Index. Downloaded from http://www.juniper.net/us/en/local/pdf/additional-resources/7100155-en.pdf, 2012.

[33] Ryan Johnson, Angelos Stavrou, and Vincent Sritapan. Improving traditional android mdms with non-traditional means. In *Technologies for Homeland Security (HST), 2016 IEEE Symposium on*, pages 1–6. IEEE, 2016.

[34] Yong Wang, Jinpeng Wei, and K. Vangury. Bring your own device security issues and challenges. In *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*, pages 80–85, January 2014. doi: 10.1109/CCNC.2014.6866552.

[35] Portnox. The importance of a nac solution introduction – the growing demand for nac solutions full coverage of top cis controls. http://www.portnox.com, 2017. Online; accessed 11.12.2019.

[36] Abdelmajid Lakbabi, Ghizlane Orhanou, and Said El Hajji. Network access control technology-proposition to contain new security challenges. *arXiv preprint arXiv:1304.0807*, 2013.

[37] Ryan Manuel. Network access control market driven by rising security issues owing to unwanted devices or unauthorized users that initiate network breaches till 2021 | million insights. https://www.prnewswire.com/news-releases/network-access-control-market-driven-by-rising-security-issues-owing-to-unwanted-devices-or-unauthorized-users-that-initiate-network-breaches-till-2021-million-insights-894728906.html, 2018. Online; accessed 11.12.2019.

[38] Ehsan Amiri, Elham Afshar, Hamid Reza Naji, and Mahdi Maleknasab Ardekani. Survey on network access control technology in manets. In *Innovation Management and Technology Research (ICIMTR), 2012 International Conference on*, pages 367–372. IEEE, 2012.

[39] Eun Byol Koh, Joohyung Oh, and Chaete Im. A study on security threats and dynamic access control technology for byod, smart-work environment. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 2, 2014.

[40] Dongwan Kang, Joohyung Oh, and Chaetae Im. Context based smart access control on byod environments. In *International Workshop on Information Security Applications*, pages 165–176. Springer, 2014.

[41] Samsung Inc. Samsung Mobile BYOD Index : Comparing IT and End User Outlooks on Bring Your Own Device. *Samsung Group*, 2013.

[42] D A Knox and T Kunz. 37 Wireless Fingerprints Inside a Wireless Sensor Network. *ACM Trans. Sensor Netw. Article*, 11(37), 2015. doi: 10.1145/2658999. URL http://dx.doi.org/10.1145/2658999.

[43] Joseph Matthews. Challenges to implementing network access control. *SANS, Information Security Reading Room*, pages 1–23, 2017.

[44] Sean Pike Robyn Westervelt, Christopher Kissel. Worldwide network access control forecast, 2018–2022: Internet of things drives nac resurgence. In *Market Forecast*, pages 1–7, MA, USA, 2018. URL https://www.idc.com/getdoc.jsp?containerId=WC20191119.

[45] Market and Market Analyst. Network Access Control Market by Product Type (Software and Hardware), Services (Consulting, Installation, and Maintenance and Support), User Type, Deployment Type, Vertical, and Region - Global Forecast to 2020. In *MarketandMarket Report*, pages 1–10, Online, 2019. URL https://www.marketsandmarkets.com/Market-Reports/network-access-control-market-133813621.html.

[46] Grand View Research. Network Access Control (NAC) Market Analysis By Type (Hardware, Software), By Service (Integration, Training, Support & Maintenance, Professional Services), By End-Use (BFSI, Government, Academia, Healthcare, Manufacturing, IT & Telecommunications) And Segment Forecasts To 2022. In *Network Access Control (NAC) Market Size Report, 2022*, pages 1–15, Online, 2016. URL https://www.grandviewresearch.com/industry-analysis/network-access-control-market.

[47] Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and Shahid Khan. Future internet: the internet of things architecture, possible applications and key challenges. In *Frontiers of Information Technology (FIT), 2012 10th International Conference on*, pages 257–260. IEEE, 2012.

[48] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

[49] P. C. Castro, J. W. Ligman, M. Pistoia, J. Ponzo, G. S. Thomas, S. P. Wood, and M. Baluda. Enabling bring-your-own-device using mobile application instrumentation. *IBM Journal of Research and Development*, 57(6):7:1–7:11, November 2013. ISSN 0018-8646. doi: 10.1147/JRD.2013.2279640.

[50] Jhih-Yan Lin, Chu-Chuan Lee, Chao-Chun Yen, Shih-Chun Hsu, Cheng-Hung Hsieh, and Chun-Hao Lin. A dynamic network access control mechanism for virtual desktop environment. In *Network Operations and Management Symposium (APNOMS), 2013 15th Asia-Pacific*, pages 1–3. IEEE, 2013.

[51] Jon Matias, Jokin Garay, Alaitz Mendiola, Nerea Toledo, and Eduardo Jacob. Flownac: Flow-based network access control. In *Software Defined Networks (EWSDN), 2014 Third European Workshop on*, pages 79–84. IEEE, 2014.

[52] Inverse Inc. PacketFence As fully supported, trusted, Free and Open Source network access control (NAC) solution, March 2013. URL https://github.com/inverse-inc/packetfence.

[53] Lee Chao. *Utilizing Open Source Tools for Online Teaching and Learning: Applying Linux Technologies: Applying Linux Technologies*. IGI Global, 2009.

[54] David Jaramillo, Borko Furht, and Ankur Agarwal. Mobile virtualization technologies. In *Virtualization Techniques for Mobile Systems*, pages 5–20. Springer, 2014.

[55] Ken Barr, Prashanth Bungale, Stephen Deasy, Viktor Gyuris, Perry Hung, Craig Newell, Harvey Tuch, and Bruno Zoppis. The vmware mobile virtualization platform: is that a hypervisor in your pocket? *ACM SIGOPS Operating Systems Review*, 44(4):124–135, 2010.

[56] Jeremy Andrus, Christoffer Dall, Alexander Van't Hof, Oren Laadan, and Jason Nieh. Cells: a virtual mobile smartphone architecture. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, pages 173–187. ACM, 2011.

[57] Yury Zhauniarovich, Giovanni Russello, Mauro Conti, Bruno Crispo, and Earlence Fernandes. Moses: supporting and enforcing security profiles on smartphones. *IEEE Transactions on Dependable and Secure Computing*, 11(3):211–223, 2014.

[58] Yin Yan, Chunyu Chen, Karthik Dantu, Steven Y Ko, and Lukasz Ziarek. Using a multi-tasking vm for mobile applications. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*, pages 93–98. ACM, 2016.

[59] Fu Chunle, He Qinggang, Wang Bailing, and Han Xixian. A communication supportable generic model for mobile vpn on android os. In *Computers and Communication (ISCC), 2016 IEEE Symposium on*, pages 1039–1046. IEEE, 2016.

[60] Weijing Chen and Keith Joseph Allen. Broadband access for virtual private networks, June 4 2019. US Patent App. 10/313,306.

[61] Sara Ali, Muhammad Nauman Qureshi, and Abdul Ghafoor Abbasi. Analysis of byod security frameworks. In *2015 Conference on Information Assurance and Cyber Security (CIACS)*, pages 56–61. IEEE, 2015.

[62] Alexander V Uskov. Information security of mobile vpn: Conceptual models and design methodology. In *Electro/Information Technology (EIT), 2012 IEEE International Conference on*, pages 1–6. IEEE, 2012.

[63] Richa Garg, Mayank Gupta, Ruhul Amin, Kalgi Patel, SK Hafizul Islam, and GP Biswas. Design of secure authentication protocol in socks v5 for vpn using mobile phone. In *Trends in Automation, Communications and Computing Technology (I-TACT-15), 2015 International Conference on*, volume 1, pages 1–6. IEEE, 2015.

[64] Morufu Olalere, Mohd Taufik Abdullah, Ramlan Mahmod, and Azizol Abdullah. A review of bring your own device on security issues. *Sage Open*, 5(2): 2158244015580372, 2015.

[65] Vanessa Frias-Martinez, Joseph Sherrick, Salvatore J Stolfo, and Angelos D Keromytis. A network access control mechanism based on behavior profiles. In *Computer Security Applications Conference, 2009. ACSAC'09. Annual*, pages 3–12. IEEE, 2009.

[66] Xuejun Song and Tiantong You. A network access control mechanism based on role and behavior. In *2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*, pages 95–99. IEEE, 2010.

[67] Luís Oliveira, Joel Rodrigues, Amaro de Sousa, and Jaime Lloret. A network access control framework for 6lowpan networks. *Sensors*, 13(1):1210–1230, 2013.

[68] Joseph Davies and Tony Northrup. *Windows Server 2008 Networking and Network Access Protection-NAP*. Microsoft Press, 2008.

[69] Craig S Etchegoyen. Network access protection, June 2 2015. US Patent 9,047,458.

[70] A Selcuk Uluagac, Sakthi V Radhakrishnan, Cherita Corbett, Antony Baca, and Raheem Beyah. A passive technique for fingerprinting wireless devices with wired-side observations. In *2013 IEEE conference on communications and network security (CNS)*, pages 305–313. IEEE, 2013.

[71] Taeeun Kim and Hwankuk Kim. A system for detection of abnormal behavior in byod based on web usage patterns. In *Information and Communication Technology Convergence (ICTC), 2015 International Conference on*, pages 1288–1293. IEEE, 2015.

[72] Manpreet Singh and Manjeet Singh Patterh. Formal specification of common criteria based access control policy model. *IJ Network Security*, 11(3):139–148, 2010.

[73] Yong Wang, Jinpeng Wei, and Karthik Vangury. Bring your own device security issues and challenges. In *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*, pages 80–85. IEEE, 2014.

[74] Denys A Flores, Farrukh Qazi, and Arshad Jhumka. Bring your own disclosure: Analysing byod threats to corporate information. In *2016 IEEE Trustcom/BigDataSE/ISPA*, pages 1008–1015. IEEE, 2016.

[75] Abubakar Bello Garba, Jocelyn Armarego, David Murray, and William Kenworthy. Review of the information security and privacy challenges in bring your own device (byod) environments. *Journal of Information Privacy and Security*, 11(1): 38–54, 2015.

[76] Jason RC Nurse, Arnau Erola, Ioannis Agrafiotis, Michael Goldsmith, and Sadie Creese. Smart insiders: exploring the threat from insiders using the internet-of-things. In *2015 International Workshop on Secure Internet of Things (SIoT)*, pages 5–14. IEEE, 2015.

[77] Donald Welch and Scott Lathrop. Wireless security threat taxonomy. In *IEEE Systems, Man and Cybernetics Society Information Assurance Workshop, 2003.*, pages 76–83. IEEE, 2003.

[78] Rupinder S Gill. *Intrusion detection techniques in wireless local area networks.* PhD thesis, Queensland University of Technology, 2009.

[79] Francisco J Aparicio-Navarro. *Using metrics from multiple layers to detect attacks in wireless networks.* PhD thesis, © Francisco Javier Aparicio Navarro, 2014.

[80] Xia Ye and Junshan Li. A security architecture based on immune agents for manet. In *2010 International Conference on Wireless Communication and Sensor Computing (ICWCSC)*, pages 1–5. IEEE, 2010.

[81] Simon LR Vrhovec. Safe use of mobile devices in the cyberspace. In *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1397–1401. IEEE, 2016.

[82] Şerif Bahtiyar, Mehmet Barış Yaman, and Can Yılmaz Altınığne. A multi-dimensional machine learning approach to predict advanced malware. *Computer Networks*, 160:118–129, 2019.

[83] Keith W Miller, Jeffrey Voas, and George F Hurlburt. Byod: Security and privacy considerations. *It Professional*, 14(5):53–55, 2012.

[84] Manmeet Mahinderjit Singh, Soh Sin Siang, Oh Ying San, Nurul Hashimah, Ahamed Hassain Malim, Azizul Rahman, and Mohd Shariff. Security attacks taxonomy on bring your own devices (BYOD) model. *International Journal of Mobile Network Communications & Telematics (IJMNCT) Vol*, 4(5):1–17, 2014.

[85] Trend Micro. Ransomware. https://www.trendmicro.com/ransomware, 2013. Online; accessed 11.10.2017.

[86] Georgeta Bordea, Lau Lap Bann, Manmeet Mahinderjit Singh, and Azman Samsudin. The third information systems international conference 2015 trusted security policies for tackling advanced persistent threat via spear phishing in byod environment. *Procedia Computer Science*, 72(113):129 – 136, 2015. ISSN 1877-0509. doi: http://dx.doi.org/10.1016/j.procs.2015.12.113. URL http://www.sciencedirect.com/science/article/pii/S1877050915035747.

[87] Ruffin White, Gianluca Caiazza, Chenxu Jiang, Xinyue Ou, Zhiyue Yang, Agostino Cortesi, and Henrik Christensen. Network reconnaissance and vulnerability excavation of secure dds systems. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 57–66. IEEE, 2019.

[88] Xuetao Wei, Nicholas C Valler, Harsha V Madhyastha, Iulian Neamtiu, and Michalis Faloutsos. A behavior-aware profiling of handheld devices. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pages 846–854. IEEE, 2015.

[89] Harish Sharma, Kannan Govindan, Ramesh C Poonia, Sandeep Kumar, and M Wael. *Advances in Computing and Intelligent Systems*. Springer, 2020.

[90] Jason Franklin, Damon McCoy, Parisa Tabriz, Vicentiu Neagoe, Jamie V Randwyk, and Douglas Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *USENIX Security Symposium*, volume 3, pages 16–89, 2006.

[91] Loh Chin Choong Desmond, Cho Chia Yuan, Tan Chung Pheng, and Ri Seng Lee. Identifying unique devices through wireless fingerprinting. In *Proceedings of the first ACM conference on Wireless network security*, pages 46–55. ACM, 2008.

[92] Ke Gao, Cherita Corbett, and Raheem Beyah. A passive approach to wireless device fingerprinting. In *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*, pages 383–392. IEEE, 2010.

[93] Suman Jana and Sneha K Kasera. On fast and accurate detection of unauthorized wireless access points using clock skews. *IEEE Transactions on Mobile Computing*, 9(3):449–462, 2009.

[94] Christoph Neumann, Olivier Heen, and Stéphane Onno. An empirical study of passive 802.11 device fingerprinting. In *2012 32nd International Conference on Distributed Computing Systems Workshops*, pages 593–602. IEEE, 2012.

[95] Chrisil Arackaparambil, Sergey Bratus, Anna Shubina, and David Kotz. On the reliability of wireless fingerprinting using clock skews. In *Proceedings of the third ACM conference on Wireless network security*, pages 169–174. ACM, 2010.

[96] Cherita Corbett, Raheem Beyah, and John Copeland. A passive approach to wireless nic identification. In *2006 IEEE International Conference on Communications*, volume 5, pages 2329–2334. IEEE, 2006.

[97] Sakthi Vignesh Radhakrishnan, A Selcuk Uluagac, and Raheem Beyah. Gtid: A technique for physical device and device type fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, 12(5):519–532, 2015.

[98] Qiang Xu, Rong Zheng, Walid Saad, and Zhu Han. Device fingerprinting in wireless networks: Challenges and opportunities. *IEEE Communications Surveys & Tutorials*, 18(1):94–104, 2015.

[99] Asish Kumar Dalai and Sanjay Kumar Jena. Wdtf: A technique for wireless device type fingerprinting. *Wireless Personal Communications*, 97(2):1911–1928, 2017.

[100] Fudong Li. *Behaviour profiling for mobile devices*. PhD thesis, University of Plymouth, 2012.

[101] Ye Zhao. Behavior-based traffic profiling based on access control information, December 31 2013. US Patent 8,621,615.

[102] Paul Giura, Ilona Murynets, Roger Piqueras Jover, and Yevgeniy Vahlis. Is it really you?: user identification via adaptive behavior fingerprinting. In *Proceedings of the 4th ACM conference on Data and application security and privacy*, pages 333–344. ACM, 2014.

[103] Hataichanok Saevanee, Nathan Clarke, and Steven Furnell. Sms linguistic profiling authentication on mobile device. In *Network and System Security (NSS), 2011 5th International Conference on*, pages 224–228. IEEE, 2011.

[104] M Ph Stoecklin, K Singh, L Koved, X Hu, SN Chari, JR Rao, P-C Cheng, M Christodorescu, R Sailer, and DL Schales. Passive security intelligence to analyze the security risks of mobile/byod activities. *IBM Journal of Research and Development*, 60(4):9–1, 2016.

[105] Fudong Li, Nathan Clarke, Maria Papadaki, and Paul Dowland. Behaviour profiling on mobile devices. In *Emerging Security Technologies (EST), 2010 International Conference on*, pages 77–82. IEEE, 2010.

[106] Jeyanthi Hall, Michel Barbeau, and Evangelos Kranakis. Anomaly-based intrusion detection using mobility profiles of public transportation users. In *WiMob'2005), IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, 2005.*, volume 2, pages 17–24. IEEE, 2005.

[107] Timothy K Buennemeyer, Theresa M Nelson, Lee M Clagett, John P Dunning, Randy C Marchany, and Joseph G Tront. Mobile device profiling and intrusion detection using smart batteries. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, pages 296–296. IEEE, 2008.

[108] Asaf Shabtai and Yuval Elovici. Applying behavioral detection on android-based devices. In *International Conference on Mobile Wireless Middleware, Operating Systems, and Applications*, pages 235–249. Springer, 2010.

[109] Vanessa Frias-Martinez, Salvatore J Stolfo, and Angelos D Keromytis. Behavior-based network access control: A proof-of-concept. In *International Conference on Information Security*, pages 175–190. Springer, 2008.

[110] Tao Qin, Xiaohong Guan, Chenxu Wang, and Zhaoli Liu. Mucm: multilevel user cluster mining based on behavior profiles for network monitoring. *IEEE Systems Journal*, 9(4):1322–1333, 2015.

[111] Ilias Tsompanidis, Ahmed H Zahran, and Cormac J Sreenan. Mobile network traffic: A user behaviour model. In *Wireless and Mobile Networking Conference (WMNC), 2014 7th IFIP*, pages 1–8. IEEE, 2014.

[112] Xiaming Chen, Siwei Qiang, Jianwen Wei, Kaida Jiang, and Yaohui Jin. Passive profiling of mobile engaging behaviours via user-end application performance assessment. *Pervasive and Mobile Computing*, 29:95–112, 2016.

[113] Ahmad Jakalan, Jian Gong, and Shangdong Liu. Profiling ip hosts based on traffic behavior. In *Communication Software and Networks (ICCSN), 2015 IEEE International Conference on*, pages 105–111. IEEE, 2015.

[114] Maria Kihl, Per Ödling, Christina Lagerstedt, and Andreas Aurelius. Traffic analysis and characterization of internet user behavior. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress on*, pages 224–231. IEEE, 2010.

[115] Ali El Attar, Rida Khatoun, and Marc Lemercier. Clustering-based anomaly detection for smartphone applications. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*, pages 1–4. IEEE, 2014.

[116] Nathan Eagle and Alex Sandy Pentland. Reality mining: sensing complex social systems. *Personal and ubiquitous computing*, 10(4):255–268, 2006.

[117] Sandeep Gupta, Attaullah Buriro, and Bruno Crispo. Demystifying authentication concepts in smartphones: Ways and types to secure access. *Mobile Information Systems*, 2018, 2018.

[118] Fudong Li, Ross Wheeler, and Nathan Clarke. An evaluation of behavioural profiling on mobile devices. In *International Conference on Human Aspects of Information Security, Privacy, and Trust*, pages 330–339. Springer, 2014.

[119] Sanae Rosen, Ashkan Nikravesh, Yihua Guo, Z Morley Mao, Feng Qian, and Subhabrata Sen. Revisiting network energy efficiency of mobile apps: Performance in the wild. In *Proceedings of the 2015 Internet Measurement Conference*, pages 339–345. ACM, 2015.

[120] Mohsen Sharifi, Somayeh Kafaie, and Omid Kashefi. A survey and taxonomy of cyber foraging of mobile devices. *IEEE Communications Surveys & Tutorials*, 14 (4):1232–1243, 2011.

[121] Kuai Xu, Zhi-Li Zhang, and Supratik Bhattacharyya. Internet traffic behavior profiling for network security monitoring. *IEEE/ACM Transactions on Networking (TON)*, 16(6):1241–1252, 2008.

[122] Liang Xie, Xinwen Zhang, Jean-Pierre Seifert, and Sencun Zhu. pbmds: a behavior-based malware detection system for cellphone devices. In *Proceedings of the third ACM conference on Wireless network security*, pages 37–48. ACM, 2010.

[123] Asif Iqbal Hajamydeen and Nur Izura Udzir. A refined filter for uhad to improve anomaly detection. *Security and Communication Networks*, 9(14):2434–2447, 2016.

[124] S Jha, L Kruger, T Kurtx, Y Lee, and A Smith. A filtering approach to anomaly and masquerade detection. *University of Wisconsin, Tech. Rep*, 2004.

[125] Xiao Yu, Lu An Tang, and Jiawei Han. Filtering and refinement: A two-stage approach for efficient and effective anomaly detection. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 617–626. IEEE, 2009.

[126] Anukool Lakhina, Mark Crovella, and Christophe Diot. Diagnosing network-wide traffic anomalies. *ACM SIGCOMM Computer Communication Review*, 34(4): 219–230, 2004.

[127] Florian Knorn and Douglas J Leith. Adaptive kalman filtering for anomaly detection in software appliances. In *INFOCOM Workshops 2008, IEEE*, pages 1–6. IEEE, 2008.

[128] Stefan-Iulian Handra and Horia Ciocârlie. Anomaly detection in data mining. hybrid approach between filtering-and-refinement and dbscan. In *Applied Computational Intelligence and Informatics (SACI), 2011 6th IEEE International Symposium on*, pages 75–83. IEEE, 2011.

[129] Basant Agarwal and Namita Mittal. Hybrid approach for detection of anomaly network traffic using data mining techniques. *Procedia Technology*, 6:996–1003, 2012.

[130] Hong Huang, Hussein Al-Azzawi, and Hajar Brani. Network traffic anomaly detection. *arXiv preprint arXiv:1402.0856*, 2014.

[131] Eduardo De la Hoz, Emiro De La Hoz, Andrés Ortiz, Julio Ortega, and Beatriz Prieto. Pca filtering and probabilistic som for network intrusion detection. *Neurocomputing*, 164:71–81, 2015.

[132] Igor Santos, Carlos Laorden, Xabier Ugarte-Pedrero, Borja Sanz, and Pablo G Bringas. Anomaly-based spam filtering. In *Security and Cryptography (SECRYPT), 2011 Proceedings of the International Conference on*, pages 5–14. IEEE, 2011.

[133] Carlos Laorden, Xabier Ugarte-Pedrero, Igor Santos, Borja Sanz, Javier Nieves, and Pablo G Bringas. Study on the effectiveness of anomaly detection for spam filtering. *Information Sciences*, 277:421–444, 2014.

[134] Eric Goodman, Joe Ingram, Shawn Martin, and Dirk Grunwald. Using bipartite anomaly features for cyber security applications. In *Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on*, pages 301–306. IEEE, 2015.

[135] Merima Kulin, Carolina Fortuna, Eli De Poorter, Dirk Deschrijver, and Ingrid Moerman. Data-driven design of intelligentwireless networks: An overview and tutorial. *Sensors (Switzerland)*, 16(6), 2016. ISSN 14248220. doi: 10.3390/s16060790.

[136] Selma Dilek, Hüseyin Çakır, and Mustafa Aydın. Applications of artificial intelligence techniques to combating cyber crimes: A review. *arXiv preprint arXiv:1502.03552*, 2015.

[137] Sumeet Dua and Xian Du. *Data Mining and Machine Learning in Cybersecurity*. Auerbach Publications, Boston, MA, USA, 1st edition, 2011. ISBN 1439839425, 9781439839423.

[138] William Stallings, Lawrie Brown, Michael D Bauer, and Arup Kumar Bhattacharjee. *Computer security: principles and practice*. Pearson Education Upper Saddle River, NJ, USA, 2012.

[139] Sakthi Vignesh Radhakrishnan, A. Selcuk Uluagac, and Raheem Beyah. GTID: A Technique for Physical Device and Device Type Fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, 2015. ISSN 15455971. doi: 10.1109/TDSC.2014.2369033.

[140] Steven CH Hoi, Jialei Wang, and Peilin Zhao. Libol: A library for online learning algorithms. *The Journal of Machine Learning Research*, 15(1):495–499, 2014.

[141] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009. ISSN 0360-0300. doi: 10.1145/1541880.1541882. URL http://doi.acm.org/10.1145/1541880.1541882.

[142] Donghwoon Kwon, Hyunjoo Kim, Jinoh Kim, Sang C Suh, Ikkyun Kim, and Kuinam J Kim. A survey of deep learning-based network anomaly detection. *Cluster Computing*, pages 1–13, 2017.

[143] Markus Goldstein and Seiichi Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.

[144] Mennatallah Amer, Markus Goldstein, and Slim Abdennadher. Enhancing one-class support vector machines for unsupervised anomaly detection. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, ODD '13, pages 8–15, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2335-2. doi: 10.1145/2500853.2500857. URL http://doi.acm.org/10.1145/2500853.2500857.

[145] Neha Soni and Amit Ganatra. Categorization of several clustering algorithms from different perspective: a review. *International Journal*, 2(8), 2012.

[146] Keyan Cao, Lingxu Shi, Guoren Wang, Donghong Han, and Mei Bai. Density-based local outlier detection on uncertain data. In *International Conference on Web-Age Information Management*, pages 67–71. Springer, 2014.

[147] Hongfu Liu, Jun Li, Yue Wu, and Yun Fu. Clustering with outlier removal. *arXiv preprint arXiv:1801.01899*, 2018.

[148] Ingo Mierswa and Michael Wurst. Information preserving multi-objective feature selection for unsupervised learning. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1545–1552. ACM, 2006.

[149] Trupti M Kodinariya and Prashant R Makwana. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95, 2013.

[150] Santhana Chaimontree, Katie Atkinson, and Frans Coenen. Best clustering configuration metrics: Towards multiagent based clustering. In *International Conference on Advanced Data Mining and Applications*, pages 48–59. Springer, 2010.

[151] Bernad Jumadi Dehotman Sitompul, Opim Salim Sitompul, and Poltak Sihombing. Enhancement clustering evaluation result of davies-bouldin index with determining initial centroid of k-means algorithm. *Journal of Physics: Conference Series*, 1235(1):012015, 2019.

[152] Junwei Xiao, Jianfeng Lu, and Xiangyu Li. Davies bouldin index based hierarchical initialization k-means. *Intelligent Data Analysis*, 21(6):1327–1338, 2017.

[153] Firman Tempola and Achmad Fuad Assagaf. Clustering of potency of shrimp in indonesia with k-means algorithm and validation of davies-bouldin index. In *International Conference on Science and Technology (ICST 2018)*. Atlantis Press, 2018.

[154] Juan Carlos Rojas Thomas, Matilde Santos Peñas, and Marco Mora. New version of davies-bouldin index for clustering validation based on cylindrical distance. In *2013 32nd International Conference of the Chilean Computer Science Society (SCCC)*, pages 49–53. IEEE, 2013.

[155] José María Luna-Romera, Jorge García-Gutiérrez, María Martínez-Ballesteros, and José C Riquelme Santos. An approach to validity indices for clustering techniques in big data. *Progress in Artificial Intelligence*, pages 1–14, 2018.

[156] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9):1641 – 1650, 2003. ISSN 0167-8655. doi: https://doi.org/10.1016/S0167-8655(03)00003-5. URL http://www.sciencedirect.com/science/article/pii/S0167865503000035.

[157] Lian Duan, Lida Xu, Ying Liu, and Jun Lee. Cluster-based outlier detection. *Annals of Operations Research*, 168(1):151–168, Apr 2009. ISSN 1572-9338. doi: 10.1007/s10479-008-0371-9. URL https://doi.org/10.1007/s10479-008-0371-9.

[158] Markus Goldstein and Seiichi Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLOS ONE*, 11(4):1–31, 04 2016. doi: 10.1371/journal.pone.0152173. URL https://doi.org/10.1371/journal.pone.0152173.

[159] Howard Rheingold. *Net smart: How to thrive online*. Mit Press, 2012.

[160] Anukrati Mehta. A comprehensive guide to types of neural networks. *Digital Vidya*, 2019.

[161] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[162] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[163] Vicente García, Ramón A Mollineda, and J Salvador Sánchez. A bias correction function for classification performance assessment in two-class imbalanced problems. *Knowledge-Based Systems*, 59:66–74, 2014.

[164] Mohammad Hossin and MN Sulaiman. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):1, 2015.

[165] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[166] Troy Raeder, George Forman, and Nitesh V Chawla. Learning from imbalanced data: Evaluation matters. In *Data mining: Foundations and intelligent paradigms*, pages 315–331. Springer, 2012.

[167] Alexei Botchkarev. Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology. *arXiv preprint arXiv:1809.03006*, 2018.

[168] Richard Lippmann, Joshua W Haines, David J Fried, Jonathan Korba, and Kumar Das. The 1999 darpa off-line intrusion detection evaluation. *Computer networks*, 34(4):579–595, 2000.

[169] Abdul Ahad ABRO, Erdal TAŞCI, and UGUR Aybars. A stacking-based ensemble learning method for outlier detection. *Balkan Journal of Electrical and Computer Engineering*, 8(2):181–185, 2020.

[170] UCSD CAIDA. Anonymized internet traces 2008 dataset, 2016.

[171] Nathan Eagle and Alex (Sandy) Pentland. CRAWDAD dataset mit/reality (v. 2005-07-01). Downloaded from https://crawdad.org/mit/reality/20050701, July 2005.

[172] A. Selcuk Uluagac. CRAWDAD dataset gatech/fingerprinting (v. 2014-06-09). Downloaded from https://crawdad.org/gatech/fingerprinting/20140609, June 2014.

[173] Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz, and Timm Euler. Yale: Rapid prototyping for complex data mining tasks. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 935–940, New York, NY, USA, 2006. ACM. ISBN 1-59593-339-5. doi: 10.1145/1150402.1150531. URL http://doi.acm.org/10.1145/1150402.1150531.

[174] Thyda Phit and Kôki Abe. Packet inter-arrival time estimation using neural network models. In *Internet Conference, Tokyo*. Citeseer, 2006.

[175] Markus Hofmann and Ralf Klinkenberg. *RapidMiner: Data mining use cases and business analytics applications*. CRC Press, 2013.

[176] Ildiko E Frank and Roberto Todeschini. *The data analysis handbook*, volume 14. Elsevier, 1994.

[177] Archana Singh, Avantika Yadav, and Ajay Rana. K-means with three different distance metrics. *International Journal of Computer Applications*, 67(10), 2013.

[178] Mathworks. Matlab, neural network toolbox 11.0. https://uk.mathworks.com/products/deep-learning.html, 2019.

[179] Mohamed Hibat Allah, Martin Ganahl, Lauren Hayward, Roger Melko, and Juan Carrasquilla. Machine learning ground states with recurrent neural network wavefunctions. *Bulletin of the American Physical Society*, 2020.

[180] Georgia Koppe, Sinan Guloksuz, Ulrich Reininghaus, and Daniel Durstewitz. Recurrent neural networks in mobile sampling and intervention. *Schizophrenia bulletin*, 45(2):272–276, 2019.

[181] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A comparative analysis of forecasting financial time series using arima, lstm, and bilstm. *arXiv preprint arXiv:1911.09512*, 2019.

[182] Maryem Rhanoui, Mounia Mikram, Siham Yousfi, and Soukaina Barzali. A cnn-bilstm model for document-level sentiment analysis. *Machine Learning and Knowledge Extraction*, 1(3):832–847, 2019.

[183] Alex Pappachen James. *Deep Learning Classifiers with Memristive Networks: Theory and Applications*, volume 14. Springer, 2019.

[184] Julien Corsini. *Analysis and evaluation of network intrusion detection methods to uncover data theft*. PhD thesis, Edinburgh Napier University, 2009.

[185] Vrushank Shah, Akshai K Aggarwal, and Nirbhay Chaubey. Performance improvement of intrusion detection with fusion of multiple sensors. *Complex & Intelligent Systems*, 3(1):33–39, 2017.

[186] Daniel B Araya, Katarina Grolinger, Hany F ElYamany, Miriam AM Capretz, and Girma Bitsuamlak. An ensemble learning framework for anomaly detection in building energy consumption. *Energy and Buildings*, 144:191–206, 2017.

[187] Damiano Bolzoni, Bruno Crispo, and Sandro Etalle. Atlantides: An architecture for alert verification in network intrusion detection systems. In *LISA*, volume 7, pages 1–12, 2007.

[188] Damiano Bolzoni, Bruno Crispo, and Sandro Etalle. Atlantides: Automatic configuration for alert verification in network intrusion detection systems. *IEEE Transactions on Electron Devices - IEEE TRANS ELECTRON DEVICES*, 01 2008.

[189] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP*, pages 108–116, 2018.

[190] Gavin E Crooks. Field guide to continuous probability distributions. *URL: http://threeplusone. com/fieldguide.(Accessed: 29.01. 2019)*, 2010.

[191] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

[192] Sullivan Frost. Network Access Control (NAC) Market, Global, Forecast to 2022 NAC Evolving As Enterprise Networks Expand Beyond Secure Walls, October 2018.

[193] Edward Amoroso. *Cyber attacks: protecting national infrastructure*. Elsevier, 2012.

Appendix A

Dataset Analysis and Device Type Profiling

A.1 K-means Clustering Results

A.1.1 Determining the number of clusters for Active network traffic Datasets

Table A.1 Active network traffic Dataset Results based on Trial and Error in Configuration settings

Device	Db_index ($k=2$)	Db_index ($k=3$)	Db_index ($k=4$)	Db_index ($k=5$)
AC1	2	6	381	458
AC2	2	6	409	482
AC3	2	6	274	423
AC4	2	6	293	437
AC5	2	6	36	2,402
AC6	2	13	287	456
AC7	2	6	17	527
AC8	2	6	512	504
AC9	2	6	424	643
A10	2	6	758	688
ACER	2	6	457	1,912
AS1	2	6	457	460
AS2	2	6	495	490
AS3	2	6	371	558
AS4	2	6	368	506
AS5	2	6	19	394
AS6	2	10	404	543
AS7	2	7	490	516
AS8	2	15	443	2,662
AS9	2	6	44	512
AS10	2	6	16	483
Asus	2	6	13	2,894
GW1	2	6	395	587
GW2	2	6	404	415
GW3	2	6	340	463
GW4	2	6	357	538
GW5	2	6	303	450
GW6	2	6	187	458
GW7	2	6	489	514
GW8	2	6	534	536
GatewayNB	2	6	458	455
G1	2	5	16	33
G2	2	6	13	67
GoogleP	2	6	14	33
L1	2	6	44	63
L2	2	37	45	79
Lenovo	2	7	44	67
T1	2	7	41	64
T2	2	7	13	35
Tablets	2	7	13	62

A.1.2 Determining the number of clusters for Isolated network traffic Datasets

Table A.2 Isolated network traffic Dataset Results based on Trial and Error in Configuration settings

Device	Db_index ($k = 2$)	Db_index ($k = 3$)	Db_index ($k = 4$)	Db_index ($k = 5$)
DN1	2	86	169	203
DN2	2	60	191	215
DN3	2	89	199	221
DN4	2	9	34	91
DN5	2	28	137	233
Dell-Netbooks	2	83	158	497
IP1	2	86	170	203
IP2	2	60	191	215
IP3	2	89	199	220
iPads	2	18	44	54
IT1	2	11	18	184
IT2	2	46	57	70
iPhone3G	2	47	151	174
IF1	2	197	356	457
IF2	2	325	367	467
iPhone4G	2	208	362	464
NP1	2	7	88	218
NP2	2	7	26	34
Nokia Phones	2	7	19	361

A.1.3 Determining the number of clusters for Passive network traffic Datasets

Table A.3 Active network traffic Dataset Results based on Trial and Error in Configuration settings

Device	Db_index ($k = 2$)	Db_index ($k = 3$)	Db_index ($k = 4$)	Db_index ($k = 5$)
AC1	2	11	24	42
AC2	2	13	31	55
AC3	2	12	20	39
AC4	2	9	42	59
AC5	2	10	21	814
AC6	2	7	53	70
AC7	2	9	22	40
AC8	2	8	17	2,882
AC9	2	9	30	42
A10	2	9	18	33
ACER	2	10	27	49
AS1	2	9	19	34
AS2	2	11	28	36
AS3	2	10	24	48
AS4	2	10	22	29
AS5	2	10	25	39
AS6	2	9	22	32
AS7	2	9	31	48
AS8	2	11	26	59
AS9	2	10	30	60
AS10	2	16	45	67
Asus	2	12	25	47
GW1	2	13	18	6,162
GW2	2	12	21	42
GW3	2	9	17	828
GW4	2	9	20	43
GW5	2	8	18	1,178
GW6	2	11	644	1,414
GW7	2	8	30	59
GW8	2	10	21	39
GatewayNB	2	10	22	971
G1	2	20	50	86
G2	2	14	39	69
GoogleP	2	16	44	73
L1	2	13	52	155
L2	2	10	247	306
Lenovo	2	15	60	201
T1	2	12	33	55
T2	2	12	23	44
Tablets	2	12	23	52

A.2 Dataset Analysis

A.2.1 Analysis of Active traffic Datasets

Table A.4 Descriptive Analysis of Ping-ICMP-Case1 Data (Real-Active Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	Min	Max	Std
AC1	C0	395,457	0.009	0.008	0.009	0.001	0.000	0.406	0.0045
	C1	33	0.990	1.004	1.007	1.011	0.518	1.022	0.087
AC2	C0	396,464	0.009	0.008	0.009	0.009	0.000	0.281	0.003
	C1	54	0.706	0.389	0.699	1.007	0.382	1.024	0.298
AC3	C0	395,847	0.009	0.008	0.009	0.009	0.000	0.396	0.005
	C1	64	0.970	1.002	1.005	1.008	0.530	1.015	0.110
AC4	C0	397,180	0.009	0.009	0.009	0.009	0.000	0.405	0.004
	C1	31	0.943	1.002	1.005	1.009	0.507	1.026	0.143
AC5	C0	396,442	0.009	0.009	0.009	0.009	0.000	1.016	0.010
	C1	1	5.744	5.744	5.744	5.744	5.744	5.744	0.0
AC6	C0	396,316	0.009	0.009	0.009	0.009	0.000	0.431	0.005
	C1	33	0.968	1.002	1.004	1.008	0.630	1.019	0.106
AC7	C0	397,669	0.009	0.009	0.009	0.009	0.000	0.217	0.003
	C1	31	0.694	0.391	0.641	1.007	0.383	1.066	0.293
AC8	C0	397,266	0.009	0.009	0.009	0.009	0.000	0.417	0.004
	C1	34	0.971	1.003	1.005	1.008	0.507	1.022	0.113
AC9	C0	398,019	0.009	0.009	0.009	0.009	0.000	0.180	0.002
	C1	56	0.696	0.389	0.695	1.0057	0.382	1.013	0.292
A10	C0	397,566	0.009	0.009	0.009	0.009	0.000	0.406	0.004
	C1	29	0.990	1.004	1.006	1.010	0.515	1.020	0.092
ACER	**C0**	**3,968,591**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**1.066**	**0.010**
	C1	**1**	**5.744**	**5.744**	**5.744**	**5.744**	**5.744**	**5.744**	**0.0**
AS1	C0	396,821	0.009	0.009	0.009	0.009	0.000	0.278	0.003
	C1	62	0.706	0.387	0.758	1.005	0.378	1.011	0.302
AS2	C0	397,570	0.009	0.009	0.009	0.009	0.000	0.393	0.004
	C1	54	0.925	0.880	1.005	1.007	0.510	1.012	0.146
AS3	C0	397,874	0.009	0.009	0.009	0.009	0.000	0.261	0.001
	C1	62	0.705	0.390	0.757	1.006	0.382	1.011	0.299
AS4	C0	397,871	0.009	0.009	0.009	0.009	0.000	0.396	0.004
	C1	34	0.963	1.002	1.006	1.008	0.519	1.021	0.131
AS5	C0	398,143	0.009	0.009	0.009	0.009	0.000	0.402	0.003
	C1	30	0.955	1.001	1.003	1.006	0.510	1.010	0.124
AS6	C0	396,873	0.009	0.009	0.009	0.009	0.000	0.404	0.005
	C1	53	0.944	0.999	1.005	1.008	0.630	1.020	0.127
AS7	C0	396,735	0.009	0.008	0.009	0.009	0.000	0.457	0.004
	C1	33	0.989	1.003	1.006	1.011	0.637	1.027	0.077
AS8	C0	395,369	0.009	0.008	0.009	0.009	0.000	1.029	0.010
	C1	1	7.070	7.070	7.070	7.070	7.070	7.070	0.0
AS9	C0	396,064	0.009	0.008	0.009	0.009	0.000	0.416	0.005
	C1	53	0.955	1.002	1.010	1.017	0.518	1.089	0.139
AS10	C0	396,108	0.009	0.009	0.009	0.009	0.000	0.433	0.005
	C1	64	0.982	1.002	1.006	1.012	0.634	1.270	0.117
Asus	**C0**	**3,969,873**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**1.270**	**0.010**
	C1	**1**	**7.0698**	**7.070**	**7.070**	**7.070**	**7.070**	**7.070**	**0.0**
GW1	C0	397,768	0.009	0.009	0.009	0.009	0.000	0.384	0.002
	C1	61	0.762	0.394	1.001	1.007	0.385	1.011	0.284
GW2	C0	397,486	0.009	0.009	0.009	0.009	0.000	0.405	0.004
	C1	31	0.947	1.001	1.005	1.008	0.50842	1.022	0.136
GW3	C0	397,048	0.009	0.009	0.009	0.009	0.000	0.40	0.004
	C1	31	0.955	1.003	1.005	1.009	0.515	1.054	0.136
GW4	C0	398,201	0.009	0.009	0.009	0.009	0.000	0.407	0.003
	C1	56	0.946	1.002	1.004	1.007	0.509	1.016	0.135
GW5	C0	396,844	0.009	0.009	0.009	0.009	0.000	0.411	0.004
	C1	55	0.985	1.002	1.006	1.009	0.506	1.012	0.095
GW6	C0	397,420	0.009	0.009	0.0009	0.009	0.000	0.399	0.004
	C1	55	0.965	1.002	1.004	1.008	0.630	1.027	0.111
GW7	C0	397,646	0.009	0.009	0.009	0.009	0.000	0.406	0.004
	C1	32	1.005	1.002	1.003	1.006	1.000	1.013	0.004
GW8	C0	397,210	0.009	0.009	0.009	0.009	0.000	0.415	0.004
	C1	36	0.938	1.000	1.004	1.008	0.507	1.019	0.140
GatewayNB	**C0**	**3,179,519**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**0.278**	**0.002**
	C1	**461**	**0.705**	**0.388**	**0.759**	**1.005**	**0.378**	**1.054**	**0.298**
G1	C0	389,003	0.009	0.009	0.009	0.009	0.000	0.011	8.847
	C1	9491	0.0138	0.012	0.013	0.014	0.011	0.086	0.003
G2	C0	385,259	0.009	0.009	0.009	0.009	0.000	0.012	0.001
	C1	13,064	0.014	0.012	0.013	0.015	0.0115	0.113	0.004
GoogleP	**C0**	**774,342**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**0.011**	**0.001**
	C1	**22,475**	**0.014**	**0.012**	**0.013**	**0.015**	**0.011**	**0.113**	**0.004**
L1	C0	399,050	0.009	0.009	0.009	0.009	0.000	0.048	0.002
	C1	78	0.087	0.087	0.094	0.098	0.048	0.125	0.020
L2	C0	399,112	0.009	0.009	0.009	0.009	0.000	0.051	0.002
	C1	69	0.095	0.093	0.096	0.099	0.053	0.121	0.010
Lenovo	**C0**	**798,163**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**0.050**	**0.002**
	C1	**146**	**0.091**	**0.091**	**0.095**	**0.099**	**0.050**	**0.125**	**0.016**
T1	C0	397,910	0.009	0.009	0.009	0.009	0.000	0.113	0.002
	C1	4	0.233	0.230	0.230	0.243	0.224	0.247	0.010
T2	C0	397,059	0.009	0.009	0.009	0.010	0.000	0.127	0.003
	C1	2	0.269	0.245	0.270	0.293	0.245	0.293	0.034
Tablets	**C0**	**765,850**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**0.013**	**0.002**
	C1	**29,125**	**0.016**	**0.013**	**0.015**	**0.017**	**0.013**	**0.293**	**0.007**

Table A.5 Descriptive Analysis of Ping-ICMP-Case2 Data (Real-Ping-ICMP-Case2 Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	Min	Max	Std
AC1	C0	393,116	0.009	0.008	0.009	0.010	0.000	0.437	0.005
	C1	61	0.935	1.001	1.006	1.014	0.509	1.046	0.157
AC2	C0	394,563	0.009	0.008	0.009	0.010	0.000	0.425	0.005
	C1	31	0.973	1.002	1.006	1.015	0.510	1.649	0.189
AC3	C0	396,468	0.009	0.009	0.009	0.009	0.000	0.289	0.002
	C1	63	0.706	0.391	0.908	1.008	0.381	1.050	0.308
AC4	C0	394861	0.009	0.008	0.009	0.010	0.000	0.426	0.005
	C1	32	0.937	1.001	1.006	1.011	0.504	1.037	0.161
AC5	C0	394,897	0.009	0.008	0.009	0.010	0.000	0.268	0.003
	C1	63	0.716	0.392	0.874	1.007	0.382	1.038	0.302
AC6	C0	396,172	0.009	0.008	0.009	0.009	0.000	0.322	0.003
	C1	53	0.701	0.394	0.661	1.006	0.383	1.019	0.292
AC7	C0	397,615	0.009	0.009	0.009	0.009	0.000	0.412	0.004
	C1	29	0.984	1.005	1.007	1.012	0.633	1.026	0.084
AC8	C0	397,360	0.009	0.009	0.009	0.009	0.000	0.356	0.002
	C1	62	0.710	0.393	0.759	1.005	0.381	1.019	0.289
AC9	C0	397,457	0.009	0.009	0.009	0.009	0.000	0.472	0.004
	C1	54	0.953	1.003	1.007	1.011	0.512	1.051	0.148
A10	C0	397,095	0.009	0.008	0.009	0.010	0.000	0.416	0.004
	C1	31	0.949	1.002	1.007	1.011	0.496	1.016	0.143
ACER	**C0**	**3,959,761**	**0.009**	**0.008**	**0.009**	**0.009**	**0.000**	**0.472**	**0.004**
	C1	**326**	**0.957**	**1.003**	**1.006**	**1.010**	**0.496**	**1.649**	**0.140**
AS1	C0	398,160	0.009	0.009	0.009	0.009	0.000	0.393	0.003
	C1	34	0.864	0.604	1.005	1.007	0.507	1.010	0.214
AS2	C0	397,087	0.009	0.009	0.009	0.009	0.000	0.427	0.004
	C1	28	1.007	1.003	1.008	1.009	1.001	1.022	0.004
AS3	C0	397,901	0.009	0.009	0.009	0.009	0.000	0.269	0.002
	C1	60	0.709	0.388	0.700	1.006	0.382	1.016	0.291
AS4	C0	397,786	0.009	0.009	0.009	0.009	0.000	0.414	0.003
	C1	34	0.969	1.002	1.004	1.007	0.510	1.023	0.112
AS5	C0	398,135	0.009	0.009	0.009	0.009	0.000	0.392	0.003
	C1	30	0.966	1.003	1.005	1.007	0.511	1.030	0.127
AS6	C0	394,642	0.009	0.008	0.009	0.010	0.000	0.449	0.005
	C1	32	0.936	0.917	1.008	1.016	0.514	1.046	0.147
AS7	C0	395,314	0.009	0.008	0.009	0.010	0.000	1.039	0.010
	C1	1	5.053	5.053	5.053	5.053	5.053	5.053	0.000
AS8	C0	395,365	0.009	0.008	0.009	0.010	0.000	0.194	0.003
	C1	65	0.712	0.390	0.808	1.007	0.384	1.041	0.298
AS9	C0	395,435	0.009	0.008	0.009	0.010	0.000	0.288	0.003
	C1	53	0.725	0.388	0.812	1.008	0.381	1.275	0.303
AS10	C0	396,678	0.009	0.009	0.009	0.009	0.000	0.265	0.003
	C1	60	0.721	0.389	1.000	1.007	0.380	1.062	0.308
Asus	**C0**	**3,966,571**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**0.481**	**0.004**
	C1	**329**	**0.967**	**1.003**	**1.006**	**1.009**	**0.494**	**5.053**	**0.266**
GW1	C0	397,674	0.009	0.009	0.009	0.009	0.000	0.401	0.004
	C1	33	0.971	1.002	1.005	1.008	0.514	1.051	0.113
GW2	C0	396,660	0.009	0.009	0.009	0.009	0.000	0.294	0.003
	C1	29	0.700	0.388	0.760	1.006	0.380	1.018	0.305
GW3	C0	396,277	0.009	0.009	0.009	0.009	0.000	0.463	0.004
	C1	55	0.990	1.002	1.006	1.009	0.507	1.027	0.093
GW4	C0	397,923	0.009	0.009	0.009	0.009	0.000	0.407	0.003
	C1	29	0.987	1.004	1.006	1.008	0.630	1.065	0.084
GW5	C0	395,006	0.009	0.009	0.009	0.009	0.000	0.412	0.004
	C1	56	0.943	1.003	1.007	1.010	0.499	1.100	0.166
GW6	C0	397,795	0.009	0.009	0.009	0.009	0.000	0.178	0.002
	C1	28	0.698	0.388	0.702	1.007	0.383	1.021	0.312
GW7	C0	397,585	0.009	0.009	0.009	0.009	0.000	0.404	0.004
	C1	34	0.966	1.002	1.004	1.008	0.508	1.013	0.113
GW8	C0	397,265	0.009	0.009	0.009	0.009	0.000	0.412	0.004
	C1	63	0.911	0.763	1.006	1.008	0.510	1.021	0.160
GatewayNB	**C0**	**3,176,050**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**0.463**	**0.004**
	C1	**463**	**0.967**	**1.003**	**1.006**	**1.008**	**0.499**	**1.100**	**0.119**
G1	C0	393,396	0.009	0.009	0.009	0.009	0.000	0.012	0.001
	C1	5834	0.015	0.013	0.014	0.017	0.012	0.084	0.004
G2	C0	384,887	0.009	0.009	0.009	0.009	0.000	0.013	0.001
	C1	12,448	0.016	0.013	0.015	0.017	0.013	0.127	0.006
GoogleP	**C0**	**778,623**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**0.012**	**0.001**
	C1	**17,942**	**0.016**	**0.013**	**0.014**	**0.017**	**0.012**	**0.127**	**0.005**
L1	C0	394,960	0.009	0.009	0.009	0.009	0.000	1.100	0.004
	C1	1	3.729	3.729	3.729	3.729	3.729	3.729	0.000
L2	C0	398,496	0.009	0.009	0.009	0.009	0.000	0.044	0.002
	C1	106	0.009	0.009	0.009	0.009	0.000	0.105	0.002
Lenovo	**C0**	**793,562**	**0.009**	**0.008**	**0.009**	**0.010**	**0.000**	**0.130**	**0.003**
	C1	**1**	**0.322**	**0.280**	**0.317**	**0.367**	**0.264**	**0.409**	**0.054**
T1	C0	397,829	0.009	0.009	0.009	0.009	0.000	0.113	0.002
	C1	5	0.233	0.230	0.230	0.243	0.224	0.247	0.010
T2	C0	396,165	0.009	0.009	0.009	0.010	0.000	0.127	0.003
	C1	5	0.269	0.245	0.270	0.293	0.245	0.293	0.034
Tablets	**C0**	**793,994**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**0.013**	**0.002**
	C1	**10**	**0.016**	**0.013**	**0.015**	**0.017**	**0.013**	**0.293**	**0.007**

A.2.2 Analysis of Isolated traffic Datasets

Table A.6 Descriptive Analysis of iPerf-Udp-Case1 Data (Isolated Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	Min	Max	Std
DN1	C0	291445	0.011	0.011	0.012	0.012	0.000	0.031	0.004
	C1	8076	0.051	0.037	0.044	0.058	0.031	0.278	0.022
DN2	C0	302380	0.011	0.012	0.012	0.012	0.000	0.044	0.002
	C1	1588	0.077	0.056	0.071	0.088	0.044	0.213	0.030
DN3	C0	305859	0.012	0.012	0.012	0.012	0.000	0.073	0.002
	C1	121	0.136	0.092	0.100	0.180	0.085	0.190	0.044
DN4	C0	224164	0.016	0.015	0.016	0.016	0.008	0.287	0.008
	C1	83271	0.001	0.000	0.001	0.001	0.000	0.008	0.001
DN5	C0	296627	0.011	0.000	0.015	0.016	0.000	0.059	0.008
	C1	2329	0.108	0.070	0.087	0.107	0.059	3.015	0.106
Dell-Netbooks	**C0**	**1507482**	**0.011**	**0.012**	**0.012**	**0.015**	**0.000**	**0.049**	**0.005**
	C1	**8378**	**0.087**	**0.057**	**0.069**	**0.092**	**0.049**	**3.015**	**0.068**
IP1	C0	291446	0.011	0.011	0.012	0.012	0.000	0.031	0.004
	C1	8075	0.051	0.037	0.044	0.058	0.031	0.278	0.022
IP2	C0	302380	0.011	0.012	0.012	0.012	0.000	0.044	0.002
	C1	1588	0.077	0.056	0.071	0.088	0.044	0.213	0.030
IP3	C0	305859	0.012	0.012	0.012	0.012	0.000	0.073	0.002
	C1	121	0.136	0.092	0.100	0.180	0.085	0.190	0.044
iPads	**C0**	**909933**	**0.012**	**0.011**	**0.012**	**0.012**	**0.000**	**1.549**	**0.013**
	C1	**6**	**4.907**	**5.001**	**5.001**	**5.001**	**4.436**	**5.002**	**0.231**
IT1	C0	2208	0.004	0.001	0.004	0.006	0.000	0.007	0.003
	C1	319446	0.011	0.011	0.011	0.011	0.007	0.042	0.001
IT2	C0	306532	0.012	0.012	0.012	0.012	0.000	0.064	0.002
	C1	250	0.122	0.119	0.122	0.125	0.067	0.135	0.005
iPhone3G	**C0**	**628186**	**0.011**	**0.011**	**0.011**	**0.012**	**0.000**	**0.064**	**0.002**
	C1	**250**	**0.122**	**0.119**	**0.122**	**0.125**	**0.067**	**0.135**	**0.005**
IF1	C0	306410	0.012	0.012	0.012	0.012	0.000	0.146	0.003
	C1	3	4.218	3.826	5.001	5.001	2.652	5.001	1.356
IF2	C0	306688	0.012	0.012	0.012	0.012	0.000	1.637	0.004
	C1	2	4.016	3.686	4.016	4.346	3.355	4.676	0.934
iPhone4G	**C0**	**613098**	**0.012**	**0.012**	**0.012**	**0.012**	**0.000**	**1.637**	**0.004**
	C1	**5**	**4.137**	**3.355**	**4.676**	**5.001**	**2.652**	**5.001**	**1.073**
NP1	C0	250221	0.006	0.001	0.003	0.011	0.000	0.019	0.005
	C1	69723	0.032	0.026	0.030	0.039	0.019	0.860	0.008
NP2	C0	228474	0.006	0.001	0.004	0.012	0.000	0.018	0.005
	C1	72450	0.031	0.025	0.030	0.038	0.018	1.650	0.010
Nokia Phones	**C0**	**478850**	**0.006**	**0.001**	**0.004**	**0.011**	**0.000**	**0.019**	**0.005**
	C1	**142018**	**0.031**	**0.026**	**0.030**	**0.039**	**0.019**	**1.650**	**0.009**

Table A.7 Descriptive Analysis of iPerf-TCP-Case2 Data (Isolated Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	Min	Max	Std
DN1	C0	840955	0.001	0.001	0.001	0.001	0.000	0.058	0.001
	C1	344	0.132	0.087	0.134	0.175	0.082	0.187	0.044
DN2	C0	1327118	0.001	0.001	0.001	0.001	0.000	0.037	0.001
	C1	320	0.074	0.052	0.061	0.082	0.038	0.284	0.039
DN3	C0	1288629	0.001	0.001	0.001	0.001	0.000	0.053	0.001
	C1	66	0.111	0.083	0.088	0.172	0.062	0.177	0.042
DN4	C0	2557115	0.001	0.000	0.001	0.001	0.000	0.004	0.000
	C1	26530	0.007	0.004	0.005	0.008	0.004	0.202	0.004
DN5	C0	3059230	0.001	0.000	0.000	0.001	0.000	0.176	0.001
	C1	17	0.428	0.268	0.447	0.543	0.233	0.657	0.154
Dell-Netbooks	**C0**	**9099736**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**0.065**	**0.001**
	C1	**588**	**0.130**	**0.086**	**0.091**	**0.175**	**0.066**	**0.657**	**0.073**
IP1	C0	840955	0.001	0.001	0.001	0.001	0.000	0.058	0.001
	C1	344	0.132	0.087	0.134	0.175	0.082	0.187	0.044
IP2	C0	1327118	0.001	0.001	0.001	0.001	0.000	0.037	0.001
	C1	320	0.074	0.052	0.061	0.082	0.038	0.284	0.039
IP3	C0	1288629	0.001	0.001	0.001	0.001	0.000	0.053	0.001
	C1	66	0.111	0.083	0.088	0.172	0.062	0.177	0.042
iPads	**C0**	**4575663**	**0.002**	**0.001**	**0.002**	**0.003**	**0.000**	**0.085**	**0.002**
	C1	**5876**	**0.168**	**0.126**	**0.152**	**0.156**	**0.085**	**2.950**	**0.139**
IT1	C0	429247	0.001	0.001	0.001	0.001	0.000	0.002	0.000
	C1	259529	0.003	0.002	0.003	0.003	0.002	0.146	0.001
IT2	C0	354579	0.002	0.001	0.001	0.002	0.000	0.004	0.001
	C1	86044	0.007	0.006	0.006	0.007	0.004	0.149	0.005
iPhone3G	**C0**	**1025156**	**0.002**	**0.001**	**0.001**	**0.002**	**0.000**	**0.004**	**0.001**
	C1	**104243**	**0.007**	**0.005**	**0.006**	**0.007**	**0.004**	**0.149**	**0.005**
IF1	C0	4162258	0.001	0.001	0.001	0.001	0.000	0.055	0.001
	C1	180	0.132	0.131	0.133	0.134	0.094	0.148	0.004
IF2	C0	4138098	0.001	0.001	0.001	0.001	0.000	0.043	0.001
	C1	228	0.133	0.132	0.134	0.135	0.069	0.149	0.005
iPhone4G	**C0**	**8300356**	**0.001**	**0.001**	**0.001**	**0.001**	**0.000**	**0.055**	**0.001**
	C1	**408**	**0.133**	**0.132**	**0.134**	**0.135**	**0.069**	**0.149**	**0.005**
NP1	C0	844462	0.001	0.001	0.001	0.001	0.000	0.791	0.006
	C1	69	1.593	0.963	1.118	2.356	0.808	5.280	0.853
NP2	C0	718428	0.001	0.001	0.001	0.001	0.000	1.395	0.010
	C1	52	2.863	2.349	2.609	2.993	1.490	10.360	1.344
Nokia Phones	**C0**	**1562927**	**0.001**	**0.001**	**0.001**	**0.001**	**0.000**	**1.316**	**0.009**
	C1	**84**	**2.660**	**2.243**	**2.538**	**2.805**	**1.335**	**10.360**	**1.176**

Table A.8 Descriptive Analysis of iPerf-UDP-Case2 Data (Isolated Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	Min	Max	Std
DN1	C0	2357977	0.001	0.001	0.001	0.001	0.000	0.026	0.001
	C1	11412	0.050	0.034	0.043	0.057	0.026	0.273	0.024
DN2	C0	2449322	0.001	0.001	0.001	0.001	0.000	0.013	0.000
	C1	16	0.032	0.026	0.032	0.036	0.020	0.043	0.007
DN3	C0	2432821	0.001	0.001	0.001	0.002	0.000	0.044	0.001
	C1	126	0.130	0.086	0.094	0.174	0.066	0.277	0.046
DN4	C0	2441282	0.001	0.000	0.001	0.001	0.000	0.339	0.003
	C1	3	4.214	3.821	5.001	5.001	2.641	5.001	1.363
DN5	C0	2417674	0.001	0.000	0.000	0.000	0.000	1.206	0.004
	C1	2	5.002	5.001	5.002	5.002	5.001	5.002	0.001
Dell-Netbooks	**C0**	**12110630**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**1.206**	**0.003**
	C1	**5**	**4.529**	**5.001**	**5.001**	**5.001**	**2.641**	**5.002**	**1.056**
IP1	C0	2357977	0.001	0.001	0.001	0.001	0.000	0.026	0.001
	C1	11412	0.050	0.034	0.043	0.057	0.026	0.273	0.024
IP2	C0	2449322	0.001	0.001	0.001	0.001	0.000	0.013	0.000
	C1	16	0.032	0.026	0.032	0.036	0.020	0.043	0.007
IP3	C0	2432821	0.001	0.001	0.001	0.002	0.000	0.044	0.001
	C1	126	0.130	0.086	0.094	0.174	0.066	0.277	0.046
iPads	**C0**	**5065600**	**0.002**	**0.001**	**0.001**	**0.002**	**0.000**	**0.083**	**0.002**
	C1	**6577**	**0.164**	**0.125**	**0.151**	**0.158**	**0.083**	**5.002**	**0.155**
IT1	C0	1817797	0.001	0.001	0.001	0.001	0.000	0.003	0.000
	C1	376014	0.004	0.003	0.004	0.005	0.003	0.124	0.002
IT2	C0	284342	0.006	0.006	0.006	0.007	0.004	0.740	0.004
	C1	1064940	0.002	0.001	0.001	0.001	0.000	0.004	0.001
iPhone3G	**C0**	**2961128**	**0.001**	**0.001**	**0.001**	**0.001**	**0.000**	**0.003**	**0.001**
	C1	**581965**	**0.006**	**0.004**	**0.006**	**0.006**	**0.003**	**0.740**	**0.003**
IF1	C0	2436508	0.001	0.001	0.002	0.002	0.000	0.210	0.001
	C1	2	4.912	4.867	4.912	4.957	4.822	5.002	0.127
IF2	C0	2434156	0.001	0.001	0.002	0.002	0.000	0.138	0.001
	C1	3	4.724	4.585	5.001	5.002	4.168	5.002	0.481
iPhone4G	**C0**	**4870664**	**0.001**	**0.001**	**0.002**	**0.002**	**0.000**	**0.210**	**0.001**
	C1	**5**	**4.799**	**4.822**	**5.001**	**5.002**	**4.168**	**5.002**	**0.361**
NP1	C0	2357053	0.002	0.001	0.001	0.002	0.000	0.162	0.002
	C1	1	4.844	4.844	4.844	4.844	4.844	4.844	0.002
NP2	C0	784694	0.002	0.001	0.001	0.002	0.000	0.060	0.845
	C1	37	59.876	60.014	60.014	60.015	54.876	60.015	0.004
Nokia Phones	**C0**	**3141748**	**0.002**	**0.001**	**0.001**	**0.002**	**0.000**	**4.844**	**0.845**
	C1	**37**	**59.876**	**60.014**	**60.014**	**60.015**	**54.876**	**60.015**	**1.176**

Table A.9 Descriptive Analysis of iPerf-UDP-Case3 Data (Isolated Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	Min	Max	Std
DN1	C0	3009408	0.001	0.001	0.001	0.001	0.000	0.002	0.000
	C1	500851	0.004	0.003	0.003	0.004	0.002	0.136	0.002
DN2	C0	4042040	0.001	0.000	0.001	0.001	0.000	0.047	0.001
	C1	941	0.093	0.070	0.086	0.102	0.047	0.416	0.038
DN3	C0	3521409	0.001	0.000	0.001	0.001	0.000	0.002	0.000
	C1	528995	0.003	0.003	0.003	0.003	0.002	0.177	0.002
DN4	C0	4805163	0.001	0.000	0.001	0.001	0.000	0.255	0.001
	C1	2	4.281	3.920	4.281	4.641	3.560	5.001	1.019
DN5	C0	5702774	0.001	0.000	0.000	0.000	0.000	0.218	0.002
	C1	2	4.097	3.646	4.097	4.549	3.194	5.001	1.278
Dell-Netbooks	**C0**	**22111581**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**0.416**	**0.001**
	C1	**4**	**4.189**	**3.469**	**4.281**	**5.001**	**3.194**	**5.001**	**0.949**
IP1	C0	3009408	0.001	0.001	0.001	0.001	0.000	0.002	0.000
	C1	500851	0.004	0.003	0.003	0.004	0.002	0.136	0.002
IP2	C0	4042040	0.001	0.000	0.001	0.001	0.000	0.047	0.001
	C1	941	0.093	0.070	0.086	0.102	0.047	0.416	0.038
IP3	C0	3521409	0.001	0.000	0.001	0.001	0.000	0.002	0.000
	C1	528995	0.003	0.003	0.003	0.003	0.002	0.177	0.002
iPads	**C0**	**5760633**	**0.002**	**0.001**	**0.001**	**0.002**	**0.000**	**0.084**	**0.002**
	C1	**6606**	**0.166**	**0.127**	**0.151**	**0.157**	**0.084**	**5.001**	**0.165**
IT1	C0	379124	0.005	0.004	0.004	0.005	0.003	0.842	0.003
	C1	1958155	0.001	0.001	0.001	0.001	0.000	0.003	0.000
IT2	C0	1071329	0.002	0.001	0.001	0.001	0.000	0.004	0.001
	C1	284944	0.006	0.006	0.006	0.006	0.004	0.743	0.004
iPhone3G	**C0**	**659471**	**0.005**	**0.004**	**0.005**	**0.006**	**0.003**	**0.842**	**0.004**
	C1	**3034081**	**0.001**	**0.001**	**0.001**	**0.001**	**0.000**	**0.003**	**0.001**
IF1	C0	4057488	0.001	0.000	0.001	0.001	0.000	1.170	0.001
	C1	2	5.001	5.001	5.001	5.001	5.001	5.001	0.000
IF2	C0	4094726	0.001	0.000	0.001	0.001	0.000	1.411	0.001
	C1	2	5.002	5.001	5.002	5.002	5.001	5.002	0.001
iPhone4G	**C0**	**8152214**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**1.411**	**0.001**
	C1	**4**	**5.001**	**5.001**	**5.001**	**5.001**	**5.001**	**5.002**	**0.001**
NP1	C0	5131698	0.001	0.000	0.001	0.001	0.000	0.122	0.001
	C1	1	4.626	4.626	4.626	4.626	4.626	4.626	0.000
NP2	C0	4206160	0.001	0.000	0.001	0.001	0.000	0.004	0.004
	C1	37923	0.006	0.005	0.006	0.006	0.004	0.116	0.001
Nokia Phones	**C0**	**9375781**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**0.122**	**0.845**
	C1	**1**	**4.626**	**4.626**	**4.626**	**4.626**	**4.626**	**4.626**	**1.176**

Table A.10 Descriptive Analysis of Ping-ICMP-Case1 Data (Isolated Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	Min	Max	Std
DN1	C0	1467593	0.001	0.000	0.001	0.001	0.000	0.002	0.000
	C1	101759	0.004	0.003	0.003	0.004	0.002	0.217	0.003
DN2	C0	1550704	0.001	0.001	0.001	0.001	0.000	0.043	0.001
	C1	305	0.086	0.054	0.072	0.096	0.044	0.698	0.054
DN3	C0	1568945	0.001	0.001	0.001	0.001	0.000	0.178	0.001
	C1	1	10.109	10.109	10.109	10.109	10.109	10.109	0.004
DN4	C0	1514332	0.001	0.001	0.001	0.001	0.000	1.968	0.004
	C1	1	10.402	10.402	10.402	10.402	10.402	10.402	0.003
DN5	C0	1514215	0.001	0.000	0.000	0.001	0.000	2.406	0.203
	C1	1	10.012	10.012	10.012	10.012	10.012	10.012	0.000
Dell-Netbooks	**C0**	**7717853**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**2.406**	**0.003**
	C1	**3**	**10.174**	**10.061**	**10.109**	**10.256**	**10.012**	**10.402**	**0.001**
IP1	C0	1467436	0.001	0.000	0.001	0.001	0.000	0.002	0.054
	C1	101916	0.004	0.003	0.003	0.004	0.002	0.217	0.001
IP2	C0	1550704	0.001	0.001	0.001	0.001	0.000	0.043	0.007
	C1	305	0.086	0.054	0.072	0.096	0.044	0.698	0.144
IP3	C0	1568945	0.001	0.001	0.001	0.001	0.000	0.178	0.002
	C1	1	10.109	10.109	10.109	10.109	10.109	10.109	0.058
iPads	**C0**	**5247176**	**0.002**	**0.001**	**0.001**	**0.002**	**0.000**	**2.065**	**0.003**
	C1	**3**	**9.951**	**9.898**	**10.009**	**10.033**	**9.786**	**10.057**	**0.003**
IT1	C0	2197441	0.002	0.001	0.001	0.002	0.000	0.178	0.001
	C1	72	0.355	0.339	0.374	0.397	0.188	0.420	0.036
IT2	C0	1599097	0.001	0.000	0.001	0.001	0.000	0.401	0.001
	C1	1	10.033	10.033	10.033	10.033	10.033	10.033	0.005
iPhone3G	**C0**	**1598943**	**0.001**	**0.001**	**0.001**	**0.001**	**0.000**	**0.697**	**0.003**
	C1	**1**	**10.075**	**10.075**	**10.075**	**10.075**	**10.075**	**10.075**	**0.030**
IF1	C0	1597996	0.001	0.000	0.001	0.001	0.000	0.188	0.001
	C1	57	0.378	0.370	0.392	0.399	0.241	0.420	0.049
IF2	C0	463984	0.002	0.001	0.001	0.002	0.000	0.004	0.001
	C1	135476	0.007	0.006	0.006	0.007	0.004	0.350	0.046
iPhone4G	**C0**	**3198040**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**0.697**	**0.001**
	C1	**2**	**10.054**	**10.044**	**10.054**	**10.065**	**10.033**	**10.075**	**0.049**
NP1	C0	1567738	0.001	0.001	0.001	0.001	0.000	0.067	0.001
	C1	3007	0.134	0.079	0.146	0.167	0.067	0.642	0.000
NP2	C0	1311670	0.001	0.001	0.001	0.001	0.000	0.052	0.004
	C1	5112	0.103	0.069	0.076	0.145	0.052	0.323	0.001
Nokia Phones	**C0**	**2879463**	**0.001**	**0.001**	**0.001**	**0.001**	**0.000**	**0.058**	**0.845**
	C1	**8064**	**0.115**	**0.072**	**0.093**	**0.159**	**0.058**	**0.642**	**1.176**

Table A.11 Descriptive Analysis of Ping-ICMP-Case2 Data (Isolated Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	Min	Max	Std
DN1	C0	1467593	0.001	0.000	0.001	0.001	0.000	0.002	0.000
	C1	101759	0.004	0.003	0.003	0.004	0.002	0.217	0.003
DN2	C0	1550704	0.001	0.001	0.001	0.001	0.000	0.043	0.001
	C1	305	0.086	0.054	0.072	0.096	0.044	0.698	0.054
DN3	C0	1568945	0.001	0.001	0.001	0.001	0.000	0.178	0.001
	C1	1	10.109	10.109	10.109	10.109	10.109	10.109	0.004
DN4	C0	1514332	0.001	0.001	0.001	0.001	0.000	1.968	0.004
	C1	1	10.402	10.402	10.402	10.402	10.402	10.402	0.003
DN5	C0	1514215	0.001	0.000	0.000	0.001	0.000	2.406	0.203
	C1	1	10.012	10.012	10.012	10.012	10.012	10.012	0.000
Dell-Netbooks	**C0**	**7717853**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**2.406**	**0.003**
	C1	**3**	**10.174**	**10.061**	**10.109**	**10.256**	**10.012**	**10.402**	**0.001**
IP1	C0	1467436	0.001	0.000	0.001	0.001	0.000	0.002	0.054
	C1	101916	0.004	0.003	0.003	0.004	0.002	0.217	0.001
IP2	C0	1550704	0.001	0.001	0.001	0.001	0.000	0.043	0.007
	C1	305	0.086	0.054	0.072	0.096	0.044	0.698	0.144
IP3	C0	1568945	0.001	0.001	0.001	0.001	0.000	0.178	0.002
	C1	1	10.109	10.109	10.109	10.109	10.109	10.109	0.058
iPads	**C0**	**5247176**	**0.002**	**0.001**	**0.001**	**0.002**	**0.000**	**2.065**	**0.003**
	C1	**3**	**9.951**	**9.898**	**10.009**	**10.033**	**9.786**	**10.057**	**0.003**
IT1	C0	2197441	0.002	0.001	0.001	0.002	0.000	0.178	0.001
	C1	72	0.355	0.339	0.374	0.397	0.188	0.420	0.036
IT2	C0	1599097	0.001	0.000	0.001	0.001	0.000	0.401	0.001
	C1	1	10.033	10.033	10.033	10.033	10.033	10.033	0.005
iPhone3G	**C0**	**1598943**	**0.001**	**0.001**	**0.001**	**0.001**	**0.000**	**0.697**	**0.003**
	C1	**1**	**10.075**	**10.075**	**10.075**	**10.075**	**10.075**	**10.075**	**0.030**
IF1	C0	1597996	0.001	0.000	0.001	0.001	0.000	0.188	0.001
	C1	57	0.378	0.370	0.392	0.399	0.241	0.420	0.049
IF2	C0	463984	0.002	0.001	0.001	0.002	0.000	0.004	0.001
	C1	135476	0.007	0.006	0.006	0.007	0.004	0.350	0.046
iPhone4G	**C0**	**3198040**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**0.697**	**0.001**
	C1	**2**	**10.054**	**10.044**	**10.054**	**10.065**	**10.033**	**10.075**	**0.049**
NP1	C0	1567738	0.001	0.001	0.001	0.001	0.000	0.067	0.001
	C1	3007	0.134	0.079	0.146	0.167	0.067	0.642	0.000
NP2	C0	1311670	0.001	0.001	0.001	0.001	0.000	0.052	0.004
	C1	5112	0.103	0.069	0.076	0.145	0.052	0.323	0.001
Nokia Phones	**C0**	**2879463**	**0.001**	**0.001**	**0.001**	**0.001**	**0.000**	**0.058**	**0.845**
	C1	**8064**	**0.115**	**0.072**	**0.093**	**0.159**	**0.058**	**0.642**	**1.176**

Table A.12 Descriptive Analysis of Scp-TCP-Case4 Data (Isolated Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	Min	Max	Std
DN1	C0	1467593	0.001	0.000	0.001	0.001	0.000	0.002	0.000
	C1	101759	0.004	0.003	0.003	0.004	0.002	0.217	0.003
DN2	C0	1550704	0.001	0.001	0.001	0.001	0.000	0.043	0.001
	C1	305	0.086	0.054	0.072	0.096	0.044	0.698	0.054
DN3	C0	1568945	0.001	0.001	0.001	0.001	0.000	0.178	0.001
	C1	1	10.109	10.109	10.109	10.109	10.109	10.109	0.004
DN4	C0	1514332	0.001	0.001	0.001	0.001	0.000	1.968	0.004
	C1	1	10.402	10.402	10.402	10.402	10.402	10.402	0.003
DN5	C0	1514215	0.001	0.000	0.000	0.001	0.000	2.406	0.203
	C1	1	10.012	10.012	10.012	10.012	10.012	10.012	0.000
Dell-Netbooks	**C0**	**7717853**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**2.406**	**0.003**
	C1	**3**	**10.174**	**10.061**	**10.109**	**10.256**	**10.012**	**10.402**	**0.001**
IP1	C0	1467436	0.001	0.000	0.001	0.001	0.000	0.002	0.054
	C1	101916	0.004	0.003	0.003	0.004	0.002	0.217	0.001
IP2	C0	1550704	0.001	0.001	0.001	0.001	0.000	0.043	0.007
	C1	305	0.086	0.054	0.072	0.096	0.044	0.698	0.144
IP3	C0	1568945	0.001	0.001	0.001	0.001	0.000	0.178	0.002
	C1	1	10.109	10.109	10.109	10.109	10.109	10.109	0.058
iPads	**C0**	**5247176**	**0.002**	**0.001**	**0.001**	**0.002**	**0.000**	**2.065**	**0.003**
	C1	**3**	**9.951**	**9.898**	**10.009**	**10.033**	**9.786**	**10.057**	**0.003**
IT1	C0	2197441	0.002	0.001	0.001	0.002	0.000	0.178	0.001
	C1	72	0.355	0.339	0.374	0.397	0.188	0.420	0.036
IT2	C0	1599097	0.001	0.000	0.001	0.001	0.000	0.401	0.001
	C1	1	10.033	10.033	10.033	10.033	10.033	10.033	0.005
iPhone3G	**C0**	**1598943**	**0.001**	**0.001**	**0.001**	**0.001**	**0.000**	**0.697**	**0.003**
	C1	**1**	**10.075**	**10.075**	**10.075**	**10.075**	**10.075**	**10.075**	**0.030**
IF1	C0	1597996	0.001	0.000	0.001	0.001	0.000	0.188	0.001
	C1	57	0.378	0.370	0.392	0.399	0.241	0.420	0.049
IF2	C0	463984	0.002	0.001	0.001	0.002	0.000	0.004	0.001
	C1	135476	0.007	0.006	0.006	0.007	0.004	0.350	0.046
iPhone4G	**C0**	**3198040**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**0.697**	**0.001**
	C1	**2**	**10.054**	**10.044**	**10.054**	**10.065**	**10.033**	**10.075**	**0.049**
NP1	C0	1567738	0.001	0.001	0.001	0.001	0.000	0.067	0.001
	C1	3007	0.134	0.079	0.146	0.167	0.067	0.642	0.000
NP2	C0	1311670	0.001	0.001	0.001	0.001	0.000	0.052	0.004
	C1	5112	0.103	0.069	0.076	0.145	0.052	0.323	0.001
Nokia Phones	**C0**	**2879463**	**0.001**	**0.001**	**0.001**	**0.001**	**0.000**	**0.058**	**0.845**
	C1	**8064**	**0.115**	**0.072**	**0.093**	**0.159**	**0.058**	**0.642**	**1.176**

A.2.3 Analysis of Passive traffic Datasets

Table A.13 Descriptive Analysis of iPerf-UDP-Case3 Data (Passive-Real Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	Min	Max	Std
L1	C0	12505346	0.000	0.000	0.000	0.000	0.000	0.001	0.000
	C1	415801	0.002	0.001	0.001	0.002	0.001	0.089	0.001
L2	C0	11181747	0.000	0.000	0.000	0.000	0.000	0.001	0.000
	C1	452340	0.002	0.001	0.002	0.002	0.001	0.574	0.002
Lenovo	**C0**	**23687261**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.001**	**0.000**
	C1	**867973**	**0.002**	**0.001**	**0.001**	**0.002**	**0.001**	**0.574**	**0.002**
T1	C0	12635525	0.000	0.000	0.000	0.000	0.000	0.129	0.000
	C1	5	1.224	1.264	1.317	1.331	0.862	1.347	0.205
T2	C0	12635525	0.000	0.000	0.000	0.000	0.000	0.129	0.000
	C1	5	1.224	1.264	1.317	1.331	0.862	1.347	0.205
Tablets	**C0**	**6897784**	**0.001**	**0.001**	**0.001**	**0.002**	**0.000**	**0.004**	**0.001**
	C1	**279794**	**0.007**	**0.005**	**0.005**	**0.007**	**0.006**	**0.312**	**0.004**

Table A.14 Descriptive Analysis of Skype-UDP-Case1 Data (Passive-Real Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	Min	Max	Std
L1	C0	384361	0.002	0.000	0.000	0.002	0.000	0.011	0.003
	C1	143127	0.020	0.015	0.019	0.023	0.011	0.114	0.007
L2	C0	382128	0.002	0.000	0.000	0.001	0.000	0.009	0.002
	C1	192799	0.016	0.012	0.016	0.019	0.009	0.116	0.004
Lenovo	**C0**	**332562**	**0.018**	**0.014**	**0.017**	**0.020**	**0.010**	**0.116**	**0.006**
	C1	**769853**	**0.002**	**0.000**	**0.000**	**0.002**	**0.000**	**0.010**	**0.003**
T1	C0	364401	0.002	0.000	0.001	0.003	0.000	0.011	0.003
	C1	142465	0.020	0.016	0.021	0.024	0.011	0.114	0.005
T2	C0	365432	0.002	0.000	0.001	0.003	0.000	0.011	0.003
	C1	141919	0.020	0.016	0.021	0.024	0.011	0.522	0.005
Tablets	**C0**	**729846**	**0.002**	**0.000**	**0.001**	**0.003**	**0.000**	**0.011**	**0.003**
	C1	**284371**	**0.020**	**0.016**	**0.021**	**0.024**	**0.011**	**0.522**	**0.005**

Table A.15 Descriptive Analysis of iPerf-TCP-Case1 Data (Passive-Real-Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	Min	Max	Std
AC1	C0	4499987	0.001	0.000	0.001	0.001	0.000	0.749	0.002
	C1	13	1.574	1.079	1.251	1.813	1.001	3.615	0.765
AC2	C0	4499979	0.001	0.000	0.001	0.001	0.000	0.421	0.001
	C1	21	0.948	0.575	0.754	1.071	0.502	2.992	0.589
AC3	C0	4499990	0.001	0.000	0.001	0.001	0.000	0.751	0.002
	C1	10	1.568	1.043	1.416	2.070	0.931	2.470	0.610
AC4	C0	4499997	0.001	0.000	0.001	0.001	0.000	1.615	0.003
	C1	3	4.129	3.437	4.149	4.831	2.724	5.514	1.395
AC5	C0	4499984	0.001	0.000	0.001	0.001	0.000	0.643	0.002
	C1	16	1.450	1.107	1.315	1.584	0.874	2.882	0.575
AC6	C0	4499985	0.001	0.000	0.001	0.001	0.000	0.649	0.001
	C1	15	1.562	1.058	1.150	1.635	0.998	4.400	0.905
AC7	C0	4499973	0.001	0.000	0.001	0.001	0.000	0.457	0.001
	C1	27	0.958	0.596	0.876	1.110	0.505	2.579	0.502
AC8	C0	4499915	0.001	0.000	0.001	0.001	0.000	0.169	0.001
	C1	85	0.341	0.253	0.256	0.377	0.179	1.328	0.195
AC9	C0	4499979	0.001	0.000	0.001	0.001	0.000	0.420	0.001
	C1	21	0.879	0.570	0.629	1.085	0.502	2.556	0.512
A10	C0	4499979	0.001	0.000	0.001	0.001	0.000	0.402	0.001
	C1	21	0.909	0.574	1.000	1.117	0.476	1.671	0.385
ACER	**C0**	**44999870**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**0.649**	**0.002**
	C1	**130**	**1.393**	**1.019**	**1.143**	**1.462**	**0.705**	**5.514**	**0.742**
AS1	C0	4499980	0.001	0.000	0.001	0.001	0.000	0.628	0.002
	C1	20	1.323	0.999	1.074	1.456	0.874	2.567	0.509
AS2	C0	4499984	0.001	0.000	0.001	0.001	0.000	0.466	0.001
	C1	16	1.090	0.967	1.038	1.187	0.567	2.194	0.374
AS3	C0	4499987	0.001	0.000	0.001	0.001	0.000	0.587	0.001
	C1	13	1.414	1.091	1.167	1.555	0.857	2.753	0.571
AS4	C0	4499990	0.001	0.000	0.001	0.001	0.000	0.570	0.001
	C1	10	1.234	0.874	1.099	1.579	0.744	2.206	0.480
AS5	C0	4499912	0.001	0.000	0.001	0.001	0.000	0.185	0.001
	C1	88	0.382	0.253	0.255	0.385	0.201	2.324	0.318
AS6	C0	4499982	0.001	0.000	0.001	0.001	0.000	0.751	0.002
	C1	18	1.710	1.087	1.368	2.249	0.873	3.834	0.835
AS7	C0	4499982	0.001	0.000	0.001	0.001	0.000	0.569	0.001
	C1	18	1.177	0.756	1.072	1.334	0.625	2.364	0.526
AS8	C0	4499986	0.001	0.000	0.001	0.001	0.000	0.632	0.002
	C1	14	1.621	1.106	1.174	1.663	0.876	3.585	0.885
AS9	C0	4499987	0.001	0.000	0.001	0.001	0.000	0.632	0.002
	C1	13	1.395	1.026	1.141	1.453	0.836	3.524	0.710
AS10	C0	4499987	0.001	0.000	0.001	0.001	0.000	1.000	0.002
	C1	13	2.579	2.019	2.393	3.032	1.404	5.522	1.070
Asus	**C0**	**44999851**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**0.632**	**0.001**
	C1	**149**	**1.454**	**0.999**	**1.149**	**1.660**	**0.730**	**5.522**	**0.759**
GW1	C0	4499992	0.001	0.000	0.001	0.001	0.000	0.505	0.001
	C1	8	1.375	1.148	1.467	1.507	1.075	1.713	0.237
GW2	C0	4499990	0.001	0.000	0.001	0.001	0.000	0.754	0.002
	C1	10	1.598	1.056	1.318	1.917	0.997	2.868	0.681
GW3	C0	4499978	0.001	0.000	0.001	0.001	0.000	0.412	0.001
	C1	22	0.971	0.628	0.977	1.278	0.504	1.757	0.361
GW4	C0	4499983	0.001	0.000	0.001	0.001	0.000	0.627	0.001
	C1	17	1.414	0.951	1.189	1.744	0.869	2.494	0.572
GW5	C0	4499964	0.001	0.000	0.001	0.001	0.000	0.313	0.001
	C1	36	0.641	0.395	0.504	0.751	0.376	1.507	0.326
GW6	C0	4499985	0.001	0.000	0.001	0.001	0.000	0.751	0.002
	C1	15	1.525	0.998	1.326	1.659	0.876	4.016	0.817
GW7	C0	4499979	0.001	0.000	0.001	0.001	0.000	0.816	0.002
	C1	21	1.633	1.003	1.164	1.951	0.873	3.885	0.834
GW8	C0	4499950	0.001	0.000	0.001	0.001	0.000	0.333	0.001
	C1	50	0.689	0.392	0.503	0.691	0.362	2.341	0.475
GatewayNB	**C0**	**35999878**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**0.644**	**0.002**
	C1	**122**	**1.341**	**0.951**	**1.151**	**1.523**	**0.681**	**4.016**	**0.622**
G1	C0	1264929	0.001	0.001	0.001	0.001	0.001	0.139	0.001
	C1	4549791	0.000	0.000	0.000	0.001	0.000	0.001	0.000
G2	C0	5039267	0.001	0.000	0.001	0.001	0.000	0.003	0.001
	C1	100560	0.005	0.003	0.004	0.005	0.003	0.173	0.003
GoogleP	**C0**	**10818131**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**0.003**	**0.000**
	C1	**136416**	**0.005**	**0.003**	**0.004**	**0.005**	**0.003**	**0.173**	**0.004**
L1	C0	4152526	0.001	0.000	0.001	0.001	0.000	0.003	0.001
	C1	108251	0.006	0.004	0.004	0.006	0.003	0.180	0.005
L2	C0	4448269	0.000	0.000	0.001	0.001	0.000	0.001	0.000
	C1	1242339	0.001	0.001	0.001	0.001	0.001	0.091	0.001
Lenovo	**C0**	**9750694**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**0.003**	**0.000**
	C1	**200691**	**0.005**	**0.003**	**0.004**	**0.005**	**0.003**	**0.180**	**0.004**
T1	C0	1162486	0.001	0.001	0.001	0.001	0.001	0.094	0.001
	C1	4469386	0.000	0.000	0.001	0.001	0.000	0.001	0.000
T2	C0	3377797	0.001	0.001	0.001	0.001	0.001	0.100	0.001
	C1	2626864	0.000	0.000	0.000	0.000	0.000	0.001	0.000
Tablets	**C0**	**9163241**	**0.000**	**0.000**	**0.000**	**0.001**	**0.000**	**0.001**	**0.000**
	C1	**2473292**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.100**	**0.001**

Table A.16 Descriptive Analysis of iPerf-UDP-Case1 Data (Passive-Real Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	Min	Max	Std
AC1	C0	4488394	0.000	0.000	0.000	0.000	0.000	0.010	0.001
	C1	11606	0.020	0.012	0.015	0.021	0.010	0.165	0.018
AC2	C0	4499783	0.000	0.000	0.000	0.000	0.000	0.060	0.001
	C1	217	0.120	0.128	0.130	0.131	0.061	0.199	0.024
AC3	C0	4499802	0.000	0.000	0.000	0.000	0.000	0.060	0.001
	C1	198	0.130	0.129	0.130	0.130	0.127	0.136	0.001
AC4	C0	4499814	0.000	0.000	0.000	0.000	0.000	0.063	0.001
	C1	186	0.127	0.129	0.130	0.131	0.065	0.144	0.013
AC5	C0	4499798	0.000	0.000	0.000	0.000	0.000	0.063	0.001
	C1	202	0.130	0.129	0.130	0.131	0.069	0.180	0.007
AC6	C0	4499812	0.000	0.000	0.000	0.000	0.000	0.055	0.001
	C1	188	0.130	0.129	0.130	0.130	0.127	0.138	0.002
AC7	C0	4499835	0.000	0.000	0.000	0.000	0.000	0.038	0.000
	C1	165	0.130	0.129	0.130	0.130	0.127	0.147	0.002
AC8	C0	4499791	0.000	0.000	0.000	0.000	0.000	0.060	0.001
	C1	209	0.121	0.128	0.129	0.130	0.061	0.141	0.020
AC9	C0	4499824	0.000	0.000	0.000	0.000	0.000	0.052	0.000
	C1	176	0.130	0.129	0.130	0.130	0.127	0.136	0.001
A10	C0	4499834	0.000	0.000	0.000	0.000	0.000	0.054	0.001
	C1	166	0.130	0.129	0.130	0.131	0.077	0.140	0.005
ACER	**C0**	**44998023**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.063**	**0.001**
	C1	**1977**	**0.126**	**0.129**	**0.130**	**0.130**	**0.063**	**0.199**	**0.015**
AS1	C0	4499834	0.000	0.000	0.000	0.000	0.000	0.061	0.001
	C1	166	0.129	0.129	0.130	0.130	0.065	0.138	0.005
AS2	C0	4499819	0.000	0.000	0.000	0.000	0.000	0.064	0.000
	C1	181	0.129	0.129	0.130	0.130	0.067	0.160	0.008
AS3	C0	4499802	0.000	0.000	0.000	0.000	0.000	0.065	0.000
	C1	198	0.130	0.129	0.130	0.130	0.128	0.137	0.001
AS4	C0	4499802	0.000	0.000	0.000	0.000	0.000	0.056	0.000
	C1	198	0.130	0.129	0.129	0.130	0.128	0.137	0.001
AS5	C0	4499824	0.000	0.000	0.000	0.000	0.000	0.039	0.000
	C1	176	0.130	0.129	0.130	0.130	0.128	0.134	0.001
AS6	C0	4499809	0.000	0.000	0.000	0.000	0.000	0.063	0.001
	C1	191	0.127	0.129	0.130	0.130	0.064	0.143	0.013
AS7	C0	4499778	0.000	0.000	0.000	0.000	0.000	0.063	0.001
	C1	222	0.128	0.129	0.130	0.131	0.065	0.148	0.012
AS8	C0	4499756	0.000	0.000	0.000	0.000	0.000	0.065	0.001
	C1	244	0.131	0.129	0.130	0.132	0.066	0.559	0.037
AS9	C0	4499816	0.000	0.000	0.000	0.000	0.000	0.064	0.001
	C1	184	0.127	0.129	0.130	0.131	0.065	0.137	0.012
AS10	C0	4499785	0.000	0.000	0.000	0.000	0.000	0.063	0.001
	C1	215	0.126	0.129	0.130	0.130	0.064	0.136	0.015
Asus	**C0**	**44998027**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.065**	**0.001**
	C1	**1973**	**0.129**	**0.129**	**0.130**	**0.130**	**0.065**	**0.559**	**0.016**
GW1	C0	4499792	0.000	0.000	0.000	0.000	0.000	0.071	0.000
	C1	208	0.146	0.129	0.130	0.130	0.074	1.931	0.143
GW2	C0	4499816	0.000	0.000	0.000	0.000	0.000	0.063	0.001
	C1	184	0.128	0.129	0.130	0.130	0.065	0.146	0.012
GW3	C0	4499814	0.000	0.000	0.000	0.000	0.000	0.064	0.001
	C1	186	0.132	0.129	0.130	0.131	0.068	0.561	0.037
GW4	C0	4499822	0.000	0.000	0.000	0.000	0.000	0.061	0.000
	C1	178	0.130	0.129	0.130	0.130	0.068	0.196	0.007
GW5	C0	4499805	0.000	0.000	0.000	0.000	0.000	0.064	0.001
	C1	195	0.129	0.129	0.130	0.130	0.065	0.562	0.042
GW6	C0	4499799	0.000	0.000	0.000	0.000	0.000	0.054	0.000
	C1	201	0.129	0.129	0.130	0.130	0.067	0.137	0.007
GW7	C0	4499802	0.000	0.000	0.000	0.000	0.000	0.057	0.000
	C1	198	0.130	0.129	0.130	0.130	0.127	0.134	0.001
GW8	C0	4499834	0.000	0.000	0.000	0.000	0.000	0.059	0.000
	C1	166	0.130	0.129	0.130	0.130	0.076	0.138	0.004
GatewayNB	**C0**	**35998483**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.066**	**0.001**
	C1	**1517**	**0.132**	**0.129**	**0.130**	**0.130**	**0.066**	**1.931**	**0.057**
G1	C0	1180814	0.001	0.001	0.001	0.001	0.001	0.246	0.001
	C1	6332616	0.000	0.000	0.000	0.000	0.000	0.001	0.000
G2	C0	8026377	0.000	0.000	0.000	0.001	0.000	0.003	0.000
	C1	58975	0.005	0.003	0.004	0.006	0.003	0.096	0.004
GoogleP	**C0**	**15512046**	**0.000**	**0.000**	**0.000**	**0.001**	**0.000**	**0.003**	**0.000**
	C1	**86736**	**0.006**	**0.004**	**0.005**	**0.007**	**0.003**	**0.246**	**0.004**
L1	C0	7568619	0.000	0.000	0.000	0.000	0.000	0.001	0.000
	C1	433437	0.002	0.001	0.001	0.002	0.001	0.092	0.002
L2	C0	7531872	0.000	0.000	0.000	0.000	0.000	0.001	0.000
	C1	420652	0.002	0.001	0.001	0.002	0.001	0.098	0.002
Lenovo	**C0**	**15100992**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.001**	**0.000**
	C1	**853588**	**0.002**	**0.001**	**0.001**	**0.002**	**0.001**	**0.098**	**0.002**
T1	C0	7847388	0.000	0.000	0.000	0.001	0.000	0.003	0.000
	C1	56983	0.006	0.004	0.005	0.007	0.003	0.085	0.003
T2	C0	7328753	0.000	0.000	0.000	0.001	0.000	0.003	0.000
	C1	107786	0.005	0.003	0.004	0.006	0.003	0.095	0.003
Tablets	**C0**	**15175346**	**0.000**	**0.000**	**0.000**	**0.001**	**0.000**	**0.003**	**0.000**
	C1	**165564**	**0.005**	**0.003**	**0.004**	**0.006**	**0.003**	**0.095**	**0.003**

Table A.17 Descriptive Analysis of iPerf-UDP-Case3 Data (Passive-Real Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	min	max	Std
AC1	C0	320879	0.011	0.011	0.011	0.011	0.000	0.071	0.003
	C1	367	0.134	0.132	0.135	0.138	0.073	0.166	0.011
AC2	C0	320974	0.011	0.011	0.011	0.011	0.000	0.070	0.003
	C1	318	0.134	0.132	0.135	0.138	0.073	0.159	0.010
AC3	C0	320865	0.011	0.011	0.011	0.011	0.000	0.072	0.003
	C1	352	0.136	0.132	0.135	0.139	0.127	0.163	0.005
AC4	C0	315	0.135	0.132	0.135	0.138	0.075	0.199	0.010
	C1	320839	0.011	0.011	0.011	0.011	0.000	0.072	0.003
AC5	C0	320878	0.011	0.011	0.011	0.011	0.000	0.072	0.003
	C1	359	0.135	0.132	0.135	0.138	0.077	0.163	0.008
AC6	C0	320966	0.011	0.011	0.011	0.011	0.000	0.068	0.002
	C1	310	0.135	0.132	0.135	0.138	0.102	0.144	0.004
AC7	C0	320906	0.011	0.011	0.011	0.011	0.000	0.065	0.002
	C1	309	0.135	0.132	0.136	0.139	0.074	0.150	0.005
AC8	C0	320916	0.011	0.011	0.011	0.011	0.000	0.069	0.002
	C1	325	0.132	0.131	0.135	0.138	0.073	0.152	0.013
AC9	C0	320917	0.011	0.011	0.011	0.011	0.000	0.061	0.002
	C1	308	0.135	0.132	0.135	0.138	0.128	0.159	0.004
A10	C0	321024	0.011	0.011	0.011	0.011	0.000	0.067	0.002
	C1	310	0.135	0.132	0.135	0.138	0.075	0.155	0.006
ACER	**C0**	**3209165**	**0.011**	**0.011**	**0.011**	**0.011**	**0.000**	**0.073**	**0.003**
	C1	**3272**	**0.135**	**0.132**	**0.135**	**0.138**	**0.073**	**0.199**	**0.008**
AS1	C0	320873	0.011	0.011	0.011	0.011	0.000	0.071	0.003
	C1	354	0.134	0.132	0.135	0.138	0.074	0.155	0.009
AS2	C0	320861	0.011	0.011	0.011	0.011	0.000	0.063	0.001
	C1	309	0.135	0.132	0.135	0.138	0.082	0.147	0.005
AS3	C0	320904	0.011	0.011	0.011	0.011	0.000	0.068	0.001
	C1	352	0.135	0.132	0.135	0.138	0.129	0.143	0.003
AS4	C0	320801	0.011	0.011	0.011	0.011	0.000	0.065	0.001
	C1	353	0.135	0.132	0.135	0.138	0.077	0.146	0.005
AS5	C0	320871	0.011	0.011	0.011	0.011	0.000	0.066	0.001
	C1	309	0.135	0.132	0.135	0.138	0.076	0.143	0.005
AS6	C0	320944	0.011	0.011	0.011	0.011	0.000	0.070	0.002
	C1	315	0.134	0.132	0.135	0.138	0.073	0.151	0.008
AS7	C0	320886	0.011	0.011	0.011	0.011	0.000	0.071	0.002
	C1	353	0.135	0.133	0.135	0.138	0.108	0.150	0.004
AS8	C0	320870	0.011	0.011	0.011	0.011	0.000	0.071	0.002
	C1	354	0.135	0.132	0.135	0.138	0.074	0.152	0.006
AS9	C0	320946	0.011	0.011	0.011	0.011	0.000	0.063	0.002
	C1	311	0.135	0.132	0.135	0.138	0.077	0.150	0.006
AS10	C0	320875	0.011	0.011	0.011	0.011	0.000	0.070	0.002
	C1	361	0.134	0.132	0.135	0.138	0.074	0.156	0.007
Asus	**C0**	**3208832**	**0.011**	**0.011**	**0.011**	**0.011**	**0.000**	**0.073**	**0.002**
	C1	**3370**	**0.135**	**0.132**	**0.135**	**0.138**	**0.074**	**0.156**	**0.006**
GW1	C0	320880	0.011	0.011	0.011	0.011	0.000	0.069	0.002
	C1	355	0.135	0.132	0.135	0.138	0.076	0.143	0.005
GW2	C0	320913	0.011	0.011	0.011	0.011	0.000	0.070	0.003
	C1	334	0.131	0.131	0.134	0.138	0.073	0.152	0.015
GW3	C0	320929	0.011	0.011	0.011	0.011	0.000	0.065	0.002
	C1	312	0.134	0.132	0.135	0.138	0.074	0.146	0.007
GW4	C0	320803	0.011	0.011	0.011	0.011	0.000	0.069	0.004
	C1	367	0.128	0.131	0.134	0.138	0.070	0.165	0.020
GW5	C0	320886	0.011	0.011	0.011	0.011	0.000	0.071	0.002
	C1	311	0.135	0.132	0.135	0.138	0.080	0.152	0.006
GW6	C0	320925	0.011	0.011	0.011	0.011	0.000	0.069	0.002
	C1	353	0.135	0.132	0.135	0.138	0.077	0.155	0.005
GW7	C0	320848	0.011	0.011	0.011	0.011	0.000	0.072	0.002
	C1	355	0.135	0.132	0.135	0.138	0.076	0.148	0.006
GW8	C0	320874	0.011	0.011	0.011	0.011	0.000	0.072	0.002
	C1	309	0.135	0.132	0.135	0.138	0.077	0.158	0.005
GatewayNB	**C0**	**2567066**	**0.011**	**0.011**	**0.011**	**0.011**	**0.000**	**0.072**	**0.002**
	C1	**2688**	**0.134**	**0.132**	**0.135**	**0.138**	**0.072**	**0.165**	**0.010**
G1	C0	317000	0.011	0.011	0.011	0.011	0.000	0.015	0.001
	C1	4143	0.020	0.016	0.017	0.020	0.015	0.295	0.008
G2	C0	311976	0.011	0.011	0.011	0.011	0.000	0.013	0.001
	C1	17096	0.015	0.013	0.014	0.015	0.013	0.080	0.003
GoogleP	**C0**	**638844**	**0.011**	**0.011**	**0.011**	**0.011**	**0.000**	**0.014**	**0.001**
	C1	**11371**	**0.018**	**0.015**	**0.016**	**0.018**	**0.014**	**0.295**	**0.006**
L1	C0	321198	0.011	0.011	0.011	0.011	0.000	0.045	0.001
	C1	61	0.093	0.091	0.093	0.096	0.052	0.101	0.006
L2	C0	321264	0.011	0.011	0.011	0.011	0.000	0.050	0.001
	C1	64	0.091	0.088	0.092	0.096	0.051	0.103	0.009
Lenovo	**C0**	**642463**	**0.011**	**0.011**	**0.011**	**0.011**	**0.000**	**0.051**	**0.001**
	C1	**124**	**0.092**	**0.090**	**0.093**	**0.096**	**0.052**	**0.103**	**0.007**
T1	C0	16866	0.015	0.013	0.014	0.015	0.013	0.073	0.002
	C1	304414	0.011	0.011	0.011	0.011	0.000	0.013	0.001
T2	C0	318213	0.011	0.011	0.011	0.011	0.000	0.016	0.001
	C1	3009	0.021	0.017	0.019	0.022	0.016	0.132	0.008
Tablets	**C0**	**623763**	**0.011**	**0.011**	**0.011**	**0.011**	**0.000**	**0.014**	**0.001**
	C1	**18739**	**0.016**	**0.014**	**0.015**	**0.017**	**0.014**	**0.132**	**0.004**

Table A.18 Descriptive Analysis of iPerf-UDP-Case4 Data (Passive-Real Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	Min	Max	Std
AC1	C0	4499987	0.001	0.001	0.001	0.001	0.000	0.062	0.001
	C1	13	0.130	0.130	0.130	0.131	0.626	0.157	0.007
AC2	C0	4499980	0.001	0.001	0.001	0.001	0.000	0.063	0.001
	C1	20	0.130	0.130	0.130	0.131	0.563	0.209	0.010
AC3	C0	4499968	0.001	0.001	0.001	0.001	0.000	0.064	0.001
	C1	32	0.130	0.130	0.130	0.131	0.544	0.141	0.001
AC4	C0	4499981	0.001	0.001	0.001	0.001	0.000	0.064	0.001
	C1	19	0.129	0.130	0.130	0.131	0.568	0.144	0.008
AC5	C0	4499979	0.001	0.001	0.001	0.001	0.000	0.064	0.001
	C1	21	0.128	0.129	0.130	0.131	0.754	0.138	0.010
AC6	C0	4499957	0.001	0.001	0.001	0.001	0.000	0.063	0.001
	C1	43	0.129	0.129	0.130	0.131	0.331	0.160	0.009
AC7	C0	4499922	0.001	0.001	0.001	0.001	0.000	0.064	0.001
	C1	78	0.130	0.129	0.130	0.131	0.227	0.157	0.005
AC8	C0	4499964	0.001	0.001	0.001	0.001	0.000	0.062	0.001
	C1	36	0.130	0.130	0.130	0.131	0.346	0.143	0.002
AC9	C0	4499998	0.001	0.001	0.001	0.001	0.000	0.066	0.001
	C1	2	0.130	0.129	0.130	0.131	0.130	0.143	0.003
A10	C0	4499913	0.001	0.001	0.001	0.001	0.000	0.056	0.001
	C1	87	0.130	0.130	0.130	0.131	0.204	0.157	0.005
ACER	**C0**	**44999998**	**0.001**	**0.001**	**0.001**	**0.001**	**0.000**	**0.064**	**0.001**
	C1	**2**	**0.130**	**0.130**	**0.130**	**0.131**	**18.498**	**0.209**	**0.007**
AS1	C0	4499951	0.001	0.001	0.001	0.001	0.000	0.065	0.001
	C1	49	0.128	0.130	0.130	0.131	0.351	0.179	0.013
AS2	C0	4499954	0.001	0.001	0.001	0.001	0.000	0.063	0.000
	C1	46	0.130	0.130	0.130	0.131	0.347	0.135	0.004
AS3	C0	4499965	0.001	0.001	0.001	0.001	0.000	0.059	0.001
	C1	35	0.130	0.130	0.130	0.130	0.330	0.140	0.004
AS4	C0	4499958	0.001	0.001	0.001	0.001	0.000	0.056	0.000
	C1	42	0.130	0.130	0.130	0.130	0.335	0.135	0.003
AS5	C0	4499983	0.001	0.001	0.001	0.001	0.000	0.063	0.001
	C1	17	0.130	0.130	0.130	0.131	0.567	0.138	0.002
AS6	C0	4499977	0.001	0.001	0.001	0.001	0.000	0.064	0.001
	C1	23	0.127	0.129	0.130	0.131	0.626	0.141	0.012
AS7	C0	4499956	0.001	0.001	0.001	0.001	0.000	0.062	0.001
	C1	44	0.130	0.130	0.130	0.131	0.323	0.144	0.004
AS8	C0	4499987	0.001	0.001	0.001	0.001	0.000	0.065	0.001
	C1	13	0.130	0.130	0.130	0.131	0.742	0.143	0.007
AS9	C0	4499978	0.001	0.001	0.001	0.001	0.000	0.064	0.001
	C1	22	0.128	0.130	0.130	0.131	0.503	0.141	0.012
AS10	C0	4499956	0.001	0.001	0.001	0.001	0.000	0.064	0.001
	C1	44	0.130	0.130	0.130	0.131	0.379	0.151	0.007
Asus	**C0**	**44999743**	**0.001**	**0.001**	**0.001**	**0.001**	**0.000**	**0.065**	**0.001**
	C1	**257**	**0.129**	**0.130**	**0.130**	**0.131**	**0.442**	**0.179**	**0.008**
GW1	C0	4499933	0.001	0.001	0.001	0.001	0.000	0.065	0.000
	C1	67	0.130	0.130	0.130	0.131	0.734	0.137	0.005
GW2	C0	4499908	0.001	0.001	0.001	0.001	0.000	0.062	0.001
	C1	92	0.123	0.129	0.130	0.131	0.220	0.147	0.019
GW3	C0	4499988	0.001	0.001	0.001	0.001	0.000	0.065	0.001
	C1	12	0.129	0.130	0.130	0.131	0.627	0.151	0.007
GW4	C0	4499913	0.001	0.001	0.001	0.001	0.000	0.009	0.001
	C1	87	0.018	0.011	0.014	0.019	0.229	0.157	0.015
GW5	C0	4499902	0.001	0.001	0.001	0.001	0.000	0.012	0.001
	C1	98	0.023	0.014	0.018	0.024	0.202	0.224	0.018
GW6	C0	4499975	0.001	0.000	0.001	0.001	0.000	0.008	0.001
	C1	25	0.014	0.009	0.012	0.015	0.503	0.171	0.013
GW7	C0	4499980	0.001	0.001	0.001	0.001	0.000	0.060	0.001
	C1	20	0.130	0.130	0.130	0.131	0.701	0.141	0.005
GW8	C0	4499960	0.001	0.001	0.001	0.001	0.000	0.064	0.001
	C1	40	0.129	0.130	0.130	0.131	0.354	0.144	0.009
GatewayNB	**C0**	**35999841**	**0.001**	**0.001**	**0.001**	**0.001**	**0.000**	**0.064**	**0.001**
	C1	**159**	**0.127**	**0.130**	**0.130**	**0.131**	**0.658**	**0.224**	**0.014**
G1	C0	516929	0.001	0.001	0.001	0.001	0.000	0.005	0.000
	C1	23008	0.008	0.006	0.007	0.008	0.068	0.073	0.005
G2	C0	2130087	0.004	0.003	0.004	0.005	0.000	0.071	0.002
	C1	91205	0.001	0.001	0.001	0.001	0.013	0.003	0.000
GoogleP	**C0**	**2735307**	**0.001**	**0.001**	**0.001**	**0.001**	**0.000**	**0.004**	**0.000**
	C1	**25922**	**0.007**	**0.005**	**0.006**	**0.008**	**0.065**	**0.073**	**0.004**
L1	C0	4241494	0.001	0.001	0.001	0.001	0.000	0.003	0.001
	C1	114112	0.006	0.004	0.005	0.006	0.003	0.127	0.004
L2	C0	4328813	0.001	0.001	0.001	0.001	0.000	0.041	0.001
	C1	131900	0.086	0.085	0.086	0.089	0.003	0.094	0.008
Lenovo	**C0**	**252638**	**0.001**	**0.001**	**0.001**	**0.001**	**0.003**	**0.004**	**0.001**
	C1	**8563681**	**0.006**	**0.004**	**0.005**	**0.006**	**0.000**	**0.127**	**0.004**
T1	C0	3424589	0.001	0.001	0.001	0.002	0.000	0.003	0.001
	C1	135049	0.006	0.004	0.005	0.006	0.006	0.093	0.003
T2	C0	3473107	0.001	0.001	0.001	0.002	0.000	0.005	0.001
	C1	144833	0.008	0.005	0.006	0.009	0.006	0.312	0.005
Tablets	**C0**	**6897784**	**0.001**	**0.001**	**0.001**	**0.002**	**0.000**	**0.004**	**0.001**
	C1	**279794**	**0.007**	**0.005**	**0.005**	**0.007**	**0.006**	**0.312**	**0.004**

Table A.19 Descriptive Analysis of Ping-ICMP-Case1 Data (Passive-Real Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	min	max	Std
AC1	C0	393855	0.009	0.009	0.009	0.009	0.000	0.069	0.003
	C1	359	0.134	0.133	0.135	0.137	0.076	0.166	0.008
AC2	C0	396376	0.009	0.009	0.009	0.009	0.000	0.066	0.001
	C1	309	0.135	0.134	0.136	0.137	0.078	0.150	0.004
AC3	C0	390799	0.009	0.009	0.009	0.009	0.000	0.071	0.003
	C1	356	0.134	0.132	0.134	0.137	0.074	0.166	0.006
AC4	C0	393827	0.009	0.009	0.009	0.009	0.000	0.070	0.002
	C1	473	0.131	0.130	0.136	0.137	0.071	0.270	0.018
AC5	C0	395389	0.009	0.009	0.009	0.009	0.000	0.061	0.001
	C1	354	0.134	0.133	0.135	0.137	0.075	0.141	0.005
AC6	C0	393468	0.009	0.009	0.009	0.009	0.000	0.066	0.002
	C1	362	0.133	0.132	0.134	0.137	0.072	0.151	0.009
AC7	C0	393658	0.009	0.009	0.009	0.009	0.000	0.070	0.002
	C1	333	0.133	0.132	0.134	0.137	0.072	0.225	0.011
AC8	C0	393839	0.009	0.009	0.009	0.009	0.000	0.066	0.002
	C1	359	0.134	0.132	0.135	0.136	0.076	0.339	0.013
AC9	C0	310	0.135	0.133	0.135	0.137	0.076	0.155	0.005
	C1	395963	0.009	0.009	0.009	0.009	0.000	0.066	0.001
A10	C0	395912	0.009	0.009	0.009	0.009	0.000	0.068	0.002
	C1	311	0.134	0.132	0.135	0.137	0.078	0.142	0.006
ACER	**C0**	**3943087**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**0.071**	**0.002**
	C1	**3525**	**0.134**	**0.132**	**0.135**	**0.137**	**0.071**	**0.339**	**0.010**
AS1	C0	387655	0.009	0.009	0.009	0.009	0.000	0.153	0.005
	C1	1	42.855	42.855	42.855	42.855	42.855	42.855	0.003
AS2	C0	394210	0.009	0.009	0.009	0.009	0.000	0.071	0.006
	C1	312	0.134	0.132	0.135	0.137	0.074	0.154	0.001
AS3	C0	396224	0.009	0.009	0.009	0.009	0.000	0.065	0.004
	C1	353	0.134	0.131	0.136	0.137	0.093	0.145	0.001
AS4	C0	396255	0.009	0.009	0.009	0.009	0.000	0.064	0.004
	C1	352	0.135	0.131	0.136	0.138	0.127	0.149	0.001
AS5	C0	396882	0.009	0.009	0.009	0.009	0.000	0.042	0.003
	C1	308	0.134	0.131	0.136	0.137	0.128	0.149	0.001
AS6	C0	396468	0.009	0.009	0.009	0.009	0.000	0.048	0.003
	C1	308	0.135	0.131	0.135	0.137	0.128	0.149	0.001
AS7	C0	396042	0.009	0.009	0.009	0.009	0.000	0.066	0.006
	C1	355	0.134	0.132	0.136	0.137	0.074	0.151	0.002
AS8	C0	394735	0.009	0.009	0.009	0.009	0.000	0.070	0.009
	C1	364	0.133	0.131	0.135	0.137	0.072	0.160	0.002
AS9	C0	395395	0.009	0.009	0.009	0.009	0.000	0.068	0.005
	C1	310	0.135	0.132	0.135	0.137	0.079	0.152	0.001
AS10	C0	395747	0.009	0.009	0.009	0.009	0.000	0.071	0.006
	C1	355	0.134	0.132	0.135	0.137	0.078	0.156	0.004
Asus	**C0**	**3952630**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**0.160**	**0.002**
	C1	**1**	**42.855**	**42.855**	**42.855**	**42.855**	**42.855**	**42.855**	**0.016**
GW1	C0	394932	0.009	0.009	0.009	0.009	0.000	0.071	0.001
	C1	385	0.134	0.133	0.136	0.137	0.072	0.309	0.003
GW2	C0	396732	0.009	0.009	0.009	0.009	0.000	0.068	0.003
	C1	308	0.135	0.134	0.136	0.138	0.129	0.146	0.021
GW3	C0	393063	0.009	0.009	0.009	0.009	0.000	0.067	0.001
	C1	374	0.126	0.130	0.134	0.136	0.068	0.182	0.003
GW4	C0	396678	0.009	0.009	0.009	0.009	0.000	0.042	0.003
	C1	308	0.135	0.134	0.136	0.137	0.129	0.140	0.026
GW5	C0	393499	0.009	0.009	0.009	0.009	0.000	0.070	0.001
	C1	373	0.131	0.131	0.134	0.137	0.074	0.390	0.007
GW6	C0	395698	0.009	0.009	0.009	0.009	0.000	0.060	0.002
	C1	313	0.134	0.133	0.135	0.137	0.076	0.144	0.006
GW7	C0	391720	0.009	0.009	0.009	0.009	0.000	0.070	0.002
	C1	356	0.135	0.133	0.135	0.137	0.078	0.166	0.007
GW8	C0	394419	0.009	0.009	0.009	0.009	0.000	0.066	0.002
	C1	357	0.134	0.133	0.135	0.137	0.072	0.163	0.014
GatewayNB	**C0**	**3156748**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**0.071**	**0.002**
	C1	**2767**	**0.133**	**0.132**	**0.135**	**0.137**	**0.071**	**0.390**	**0.002**
G1	C0	45188	0.022	0.021	0.022	0.023	0.017	0.084	0.002
	C1	155603	0.011	0.010	0.010	0.012	0.000	0.017	0.004
G2	C0	163681	0.010	0.010	0.010	0.011	0.000	0.013	0.002
	C1	118665	0.016	0.014	0.016	0.017	0.013	0.111	0.004
GoogleP	**C0**	**351056**	**0.011**	**0.010**	**0.010**	**0.012**	**0.000**	**0.015**	**0.001**
	C1	**132081**	**0.019**	**0.016**	**0.018**	**0.021**	**0.015**	**0.111**	**0.003**
L1	C0	399323	0.009	0.009	0.009	0.009	0.000	0.049	0.001
	C1	61	0.094	0.092	0.094	0.096	0.084	0.100	0.003
L2	C0	399557	0.009	0.009	0.009	0.009	0.000	0.047	0.001
	C1	60	0.094	0.092	0.095	0.096	0.087	0.101	0.003
Lenovo	**C0**	**798880**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**0.049**	**0.001**
	C1	**121**	**0.094**	**0.092**	**0.094**	**0.096**	**0.084**	**0.101**	**0.004**
T1	C0	333549	0.010	0.010	0.010	0.010	0.000	0.013	0.001
	C1	17186	0.016	0.014	0.015	0.017	0.013	0.133	0.004
T2	C0	338981	0.010	0.010	0.010	0.010	0.000	0.013	0.001
	C1	13499	0.016	0.014	0.015	0.017	0.013	0.101	0.004
Tablets	**C0**	**672556**	**0.010**	**0.010**	**0.010**	**0.010**	**0.000**	**0.013**	**0.001**
	C1	**30659**	**0.016**	**0.014**	**0.015**	**0.017**	**0.013**	**0.133**	**0.004**

Table A.20 Descriptive Analysis of Ping-ICMP-Case2 Data (Passive-Real Testbed)

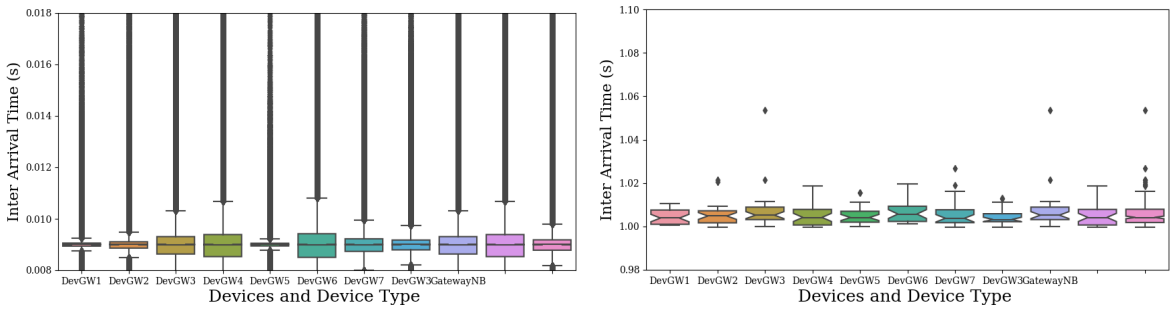
Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	Min	Max	Std
AC1	C0	390050	0.009	0.009	0.009	0.009	0.000	0.070	0.003
	C1	379	0.132	0.131	0.134	0.136	0.071	0.191	0.014
AC2	C0	395114	0.009	0.009	0.009	0.009	0.000	0.066	0.002
	C1	309	0.134	0.131	0.133	0.136	0.127	0.145	0.003
AC3	C0	389878	0.009	0.009	0.009	0.009	0.000	0.070	0.003
	C1	359	0.134	0.131	0.134	0.137	0.079	0.199	0.008
AC4	C0	395048	0.009	0.009	0.009	0.009	0.000	0.065	0.002
	C1	308	0.134	0.130	0.133	0.137	0.128	0.143	0.003
AC5	C0	355	0.133	0.131	0.134	0.136	0.072	0.148	0.005
	C1	394268	0.009	0.009	0.009	0.009	0.000	0.066	0.002
AC6	C0	392329	0.009	0.009	0.009	0.009	0.000	0.068	0.002
	C1	371	0.132	0.131	0.134	0.136	0.072	0.160	0.011
AC7	C0	392541	0.009	0.009	0.009	0.009	0.000	0.068	0.002
	C1	330	0.131	0.131	0.134	0.137	0.074	0.149	0.013
AC8	C0	392976	0.009	0.009	0.009	0.009	0.000	0.064	0.002
	C1	356	0.134	0.132	0.134	0.137	0.077	0.155	0.007
AC9	C0	395696	0.009	0.009	0.009	0.009	0.000	0.058	0.001
	C1	308	0.133	0.130	0.133	0.136	0.126	0.142	0.003
A10	C0	394008	0.009	0.009	0.009	0.009	0.000	0.059	0.002
	C1	309	0.134	0.132	0.134	0.137	0.093	0.145	0.004
ACER	**C0**	**3931910**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**0.071**	**0.002**
	C1	**3382**	**0.133**	**0.131**	**0.134**	**0.136**	**0.071**	**0.199**	**0.008**
AS1	C0	392956	0.009	0.009	0.009	0.009	0.000	0.056	0.003
	C1	355	0.134	0.131	0.134	0.137	0.076	0.150	0.006
AS2	C0	394141	0.009	0.009	0.009	0.009	0.000	0.069	0.002
	C1	310	0.134	0.131	0.134	0.137	0.111	0.159	0.004
AS3	C0	394593	0.009	0.009	0.009	0.009	0.000	0.066	0.001
	C1	354	0.134	0.131	0.134	0.136	0.082	0.149	0.005
AS4	C0	395628	0.009	0.009	0.009	0.009	0.000	0.054	0.001
	C1	352	0.134	0.131	0.134	0.136	0.129	0.143	0.003
AS5	C0	396260	0.009	0.009	0.009	0.009	0.000	0.065	0.001
	C1	308	0.134	0.131	0.134	0.136	0.129	0.147	0.003
AS6	C0	394808	0.009	0.009	0.009	0.009	0.000	0.067	0.001
	C1	313	0.133	0.131	0.134	0.136	0.076	0.156	0.008
AS7	C0	394506	0.009	0.009	0.009	0.009	0.000	0.071	0.001
	C1	354	0.134	0.131	0.135	0.137	0.073	0.152	0.006
AS8	C0	389350	0.009	0.009	0.009	0.009	0.000	0.071	0.002
	C1	361	0.134	0.131	0.134	0.137	0.072	0.164	0.007
AS9	C0	392935	0.009	0.009	0.009	0.009	0.000	0.066	0.001
	C1	309	0.134	0.131	0.134	0.136	0.112	0.146	0.004
AS10	C0	394	0.129	0.131	0.133	0.136	0.070	0.177	0.016
	C1	388268	0.009	0.009	0.009	0.009	0.000	0.069	0.003
Asus	**C0**	**3933445**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**0.071**	**0.002**
	C1	**3410**	**0.133**	**0.131**	**0.134**	**0.136**	**0.071**	**0.177**	**0.008**
GW1	C0	386684	0.009	0.009	0.009	0.009	0.000	0.061	0.002
	C1	354	0.134	0.131	0.134	0.136	0.073	0.178	0.007
GW2	C0	396131	0.009	0.009	0.009	0.009	0.000	0.051	0.001
	C1	308	0.133	0.130	0.134	0.136	0.128	0.143	0.003
GW3	C0	394429	0.009	0.009	0.009	0.009	0.000	0.067	0.002
	C1	330	0.131	0.131	0.134	0.136	0.071	0.158	0.014
GW4	C0	396168	0.009	0.009	0.009	0.009	0.000	0.052	0.001
	C1	309	0.133	0.130	0.133	0.136	0.113	0.144	0.003
GW5	C0	392477	0.009	0.009	0.009	0.009	0.000	0.071	0.003
	C1	316	0.133	0.131	0.134	0.137	0.072	0.178	0.010
GW6	C0	393993	0.009	0.009	0.009	0.009	0.000	0.064	0.002
	C1	309	0.134	0.131	0.134	0.136	0.086	0.147	0.004
GW7	C0	394472	0.009	0.009	0.009	0.009	0.000	0.057	0.001
	C1	353	0.134	0.131	0.134	0.136	0.079	0.149	0.004
GW8	C0	393613	0.009	0.009	0.009	0.009	0.000	0.070	0.001
	C1	362	0.134	0.131	0.134	0.136	0.071	0.233	0.010
GatewayNB	**C0**	**3147968**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**0.071**	**0.002**
	C1	**2640**	**0.133**	**0.131**	**0.134**	**0.136**	**0.071**	**0.233**	**0.008**
G1	C0	199437	0.011	0.010	0.010	0.012	0.000	0.016	0.002
	C1	64181	0.022	0.021	0.022	0.023	0.016	0.093	0.002
G2	C0	295996	0.012	0.010	0.011	0.014	0.000	0.053	0.003
	C1	449	0.094	0.092	0.096	0.100	0.053	0.119	0.013
GoogleP	**C0**	**444568**	**0.011**	**0.010**	**0.010**	**0.012**	**0.000**	**0.016**	**0.002**
	C1	**115495**	**0.020**	**0.017**	**0.020**	**0.022**	**0.016**	**0.119**	**0.006**
L1	C0	397811	0.009	0.009	0.009	0.009	0.000	0.049	0.001
	C1	74	0.090	0.091	0.094	0.097	0.050	0.108	0.013
L2	C0	399349	0.009	0.009	0.009	0.009	0.000	0.048	0.001
	C1	62	0.093	0.091	0.094	0.097	0.058	0.106	0.007
Lenovo	**C0**	**797161**	**0.009**	**0.009**	**0.009**	**0.009**	**0.000**	**0.050**	**0.001**
	C1	**135**	**0.092**	**0.091**	**0.094**	**0.097**	**0.054**	**0.108**	**0.010**
T1	C0	325781	0.010	0.010	0.010	0.010	0.000	0.013	0.001
	C1	21973	0.017	0.014	0.015	0.018	0.013	0.128	0.005
T2	C0	331384	0.010	0.010	0.010	0.010	0.000	0.014	0.001
	C1	15841	0.018	0.015	0.016	0.019	0.014	0.136	0.006
Tablets	**C0**	**657998**	**0.010**	**0.010**	**0.010**	**0.010**	**0.000**	**0.014**	**0.001**
	C1	**36981**	**0.017**	**0.015**	**0.016**	**0.018**	**0.014**	**0.136**	**0.005**

Table A.21 Descriptive Analysis of SCP-TCP-Case1 Data (Passive-Real Testbed)

Device	Cluster	IAT Points	Centroid	Q1	Median	Q3	Min	Max	Std
AC1	C0	4499987	0.001	0.000	0.001	0.001	0.000	0.5042	0.001
	C1	13	1.225	1.015	1.050	1.354	0.626	2.231	0.460
AC2	C0	4499980	0.001	0.000	0.001	0.001	0.000	0.50451	0.001
	C1	20	1.048	0.635	1.055	1.129	0.563	2.6042	0.540
AC3	C0	4499968	0.001	0.000	0.001	0.001	0.000	0.50593	0.002
	C1	32	1.067	0.633	0.821	1.291	0.544	2.4095	0.557
AC4	C0	4499981	0.001	0.000	0.001	0.001	0.000	0.53599	0.001
	C1	19	1.082	0.772	1.022	1.282	0.568	1.8765	0.390
AC5	C0	4499979	0.001	0.000	0.001	0.001	0.000	0.64123	0.002
	C1	21	1.359	0.913	1.006	1.393	0.754	5.9587	1.087
AC6	C0	4499957	0.001	0.000	0.001	0.001	0.000	0.29274	0.001
	C1	43	0.619	0.379	0.417	0.942	0.331	1.3262	0.320
AC7	C0	4499922	0.001	0.000	0.001	0.001	0.000	0.21856	0.001
	C1	78	0.442	0.255	0.376	0.504	0.227	1.3329	0.264
AC8	C0	4499964	0.001	0.000	0.001	0.001	0.000	0.31924	0.001
	C1	36	0.647	0.399	0.504	0.875	0.346	1.4488	0.315
AC9	C0	4499998	0.001	0.000	0.001	0.001	0.000	9.2488	0.006
	C1	2	27.747	23.122	27.747	32.371	18.498	36.995	13.079
A10	C0	4499913	0.001	0.000	0.001	0.001	0.000	0.19717	0.001
	C1	87	0.402	0.252	0.257	0.395	0.204	1.8477	0.300
ACER	**C0**	**44999998**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**9.2488**	**0.003**
	C1	**2**	**27.747**	**23.122**	**27.747**	**32.371**	**18.498**	**36.995**	**13.079**
AS1	C0	4499951	0.001	0.000	0.001	0.001	0.000	0.32888	0.001
	C1	49	0.685	0.402	0.504	0.876	0.351	1.8269	0.397
AS2	C0	4499954	0.001	0.000	0.001	0.001	0.000	0.27271	0.001
	C1	46	0.672	0.380	0.503	0.986	0.347	1.6494	0.354
AS3	C0	4499965	0.001	0.000	0.001	0.001	0.000	0.31731	0.001
	C1	35	0.660	0.379	0.501	0.782	0.330	1.86	0.411
AS4	C0	4499958	0.001	0.000	0.001	0.001	0.000	0.30542	0.001
	C1	42	0.652	0.397	0.510	0.919	0.335	1.2237	0.279
AS5	C0	4499983	0.001	0.000	0.001	0.001	0.000	0.50197	0.001
	C1	17	1.091	0.999	1.077	1.224	0.567	1.6879	0.287
AS6	C0	4499977	0.001	0.000	0.001	0.001	0.000	0.5066	0.001
	C1	23	1.171	0.812	1.037	1.208	0.626	3.6125	0.649
AS7	C0	4499956	0.001	0.000	0.001	0.001	0.000	0.29583	0.001
	C1	44	0.641	0.379	0.442	0.709	0.323	3.1397	0.472
AS8	C0	4499987	0.001	0.000	0.001	0.001	0.000	0.5027	0.001
	C1	13	1.173	0.875	1.008	1.450	0.742	2.5864	0.497
AS9	C0	4499978	0.001	0.000	0.001	0.001	0.000	0.38062	0.001
	C1	22	0.972	0.572	0.908	1.102	0.503	1.8692	0.395
AS10	C0	4499956	0.001	0.000	0.001	0.001	0.000	0.32605	0.001
	C1	44	0.666	0.425	0.503	0.768	0.379	2.1481	0.371
Asus	**C0**	**44999743**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**0.43852**	**0.001**
	C1	**257**	**0.881**	**0.504**	**0.802**	**1.077**	**0.442**	**3.6125**	**0.441**
GW1	C0	4499933	0.001	0.000	0.001	0.001	0.000	0.61686	0.002
	C1	67	1.458	1.178	1.483	1.779	0.734	2.0251	0.350
GW2	C0	4499908	0.001	0.000	0.001	0.001	0.000	0.21263	0.001
	C1	92	0.429	0.255	0.300	0.504	0.220	1.6322	0.285
GW3	C0	4499988	0.001	0.000	0.001	0.001	0.000	0.50622	0.001
	C1	12	1.170	0.897	1.115	1.270	0.627	2.4137	0.480
GW4	C0	4499913	0.001	0.000	0.001	0.001	0.000	0.21447	0.001
	C1	87	0.441	0.255	0.379	0.503	0.229	1.5906	0.259
GW5	C0	4499902	0.001	0.000	0.001	0.001	0.000	0.19436	0.001
	C1	98	0.397	0.253	0.258	0.437	0.202	1.6819	0.292
GW6	C0	4499975	0.001	0.000	0.001	0.001	0.000	0.4482	0.001
	C1	25	0.929	0.622	0.893	1.110	0.503	1.7021	0.353
GW7	C0	4499980	0.001	0.000	0.001	0.001	0.000	0.63389	0.002
	C1	20	1.316	1.004	1.143	1.505	0.701	2.4694	0.507
GW8	C0	4499960	0.001	0.000	0.001	0.001	0.000	0.32302	0.001
	C1	40	0.690	0.384	0.510	0.755	0.354	2.7952	0.485
GatewayNB	**C0**	**35999841**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**0.63612**	**0.002**
	C1	**159**	**1.294**	**1.017**	**1.178**	**1.584**	**0.658**	**2.7952**	**0.411**
G1	C0	516929	0.001	0.000	0.001	0.001	0.000	0.067616	0.002
	C1	23008	0.134	0.122	0.129	0.134	0.068	0.44301	0.026
G2	C0	2130087	0.001	0.000	0.001	0.001	0.000	0.012561	0.001
	C1	91205	0.024	0.020	0.021	0.024	0.013	2.0048	0.016
GoogleP	**C0**	**2735307**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**0.065175**	**0.004**
	C1	**25922**	**0.129**	**0.118**	**0.128**	**0.134**	**0.065**	**2.0048**	**0.033**
L1	C0	4241494	0.001	0.000	0.001	0.001	0.000	0.002817	0.000
	C1	114112	0.005	0.003	0.004	0.005	0.003	0.15524	0.004
L2	C0	4328813	0.001	0.000	0.001	0.001	0.000	0.002503	0.000
	C1	131900	0.004	0.003	0.003	0.005	0.003	0.09638	0.003
Lenovo	**C0**	**252638**	**0.005**	**0.003**	**0.004**	**0.005**	**0.003**	**0.15524**	**0.003**
	C1	**8563681**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**0.002627**	**0.000**
T1	C0	3424589	0.001	0.000	0.001	0.001	0.000	0.005725	0.001
	C1	135049	0.011	0.008	0.010	0.012	0.006	0.32368	0.004
T2	C0	3473107	0.001	0.000	0.001	0.001	0.000	0.005689	0.001
	C1	144833	0.011	0.009	0.010	0.012	0.006	0.14789	0.004
Tablets	**C0**	**6897784**	**0.001**	**0.000**	**0.001**	**0.001**	**0.000**	**0.005708**	**0.001**
	C1	**279794**	**0.011**	**0.009**	**0.010**	**0.012**	**0.006**	**0.32368**	**0.004**

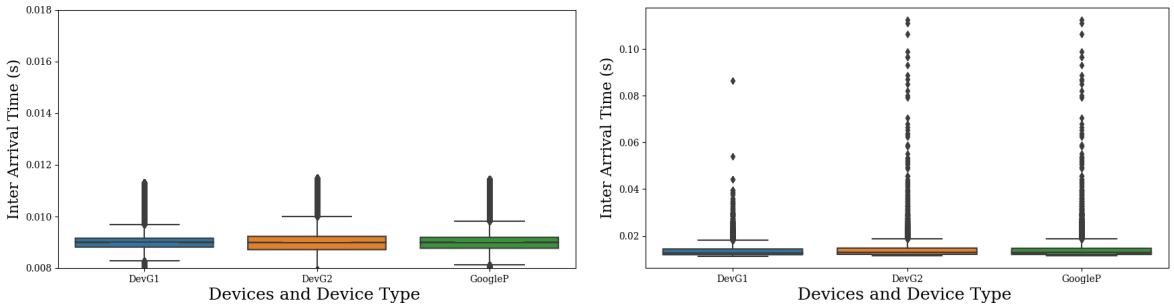
A.2.4 Notched Box plots of the datasets

Active Datasets



(a) Normal (b) Abnormal

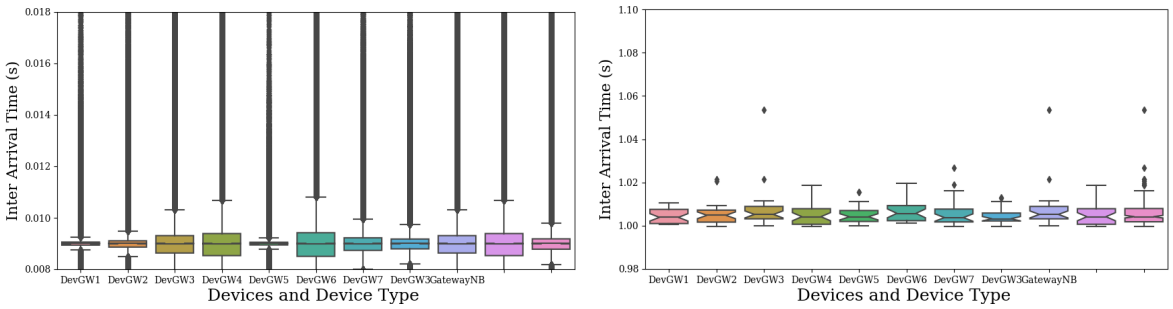
Fig. A.1 The notched box plots of normal and abnormal cluster centroid points for devices (GW1-8) and their device type (Gateway Netbooks) in Active network traffic datasets



(a) Normal (b) Abnormal

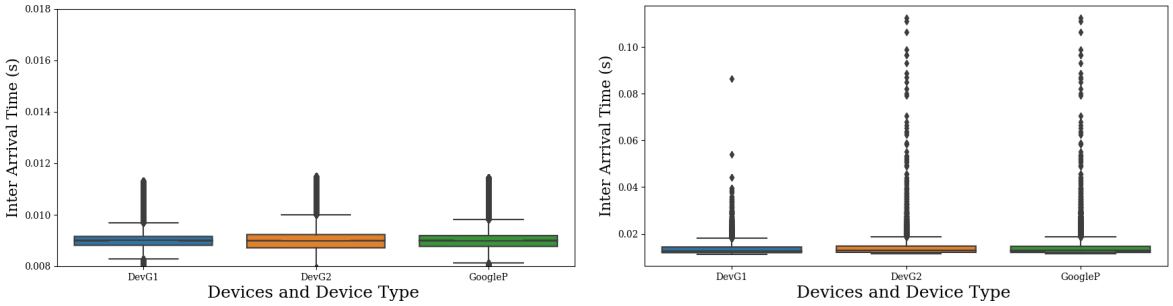
Fig. A.2 The notched box plots of normal and abnormal cluster centroid points for devices (G1-2) and their device type (Google Phones) in Active network traffic datasets

Isolated



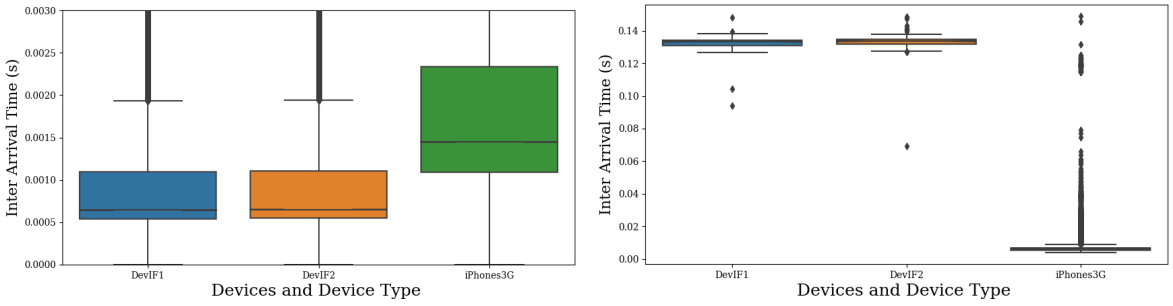
(a) Normal (b) Abnormal

Fig. A.3 The notched box plots of normal and abnormal cluster centroid points for devices (L1-2) and their device type (Lenovo Laptop) in Active network traffic datasets



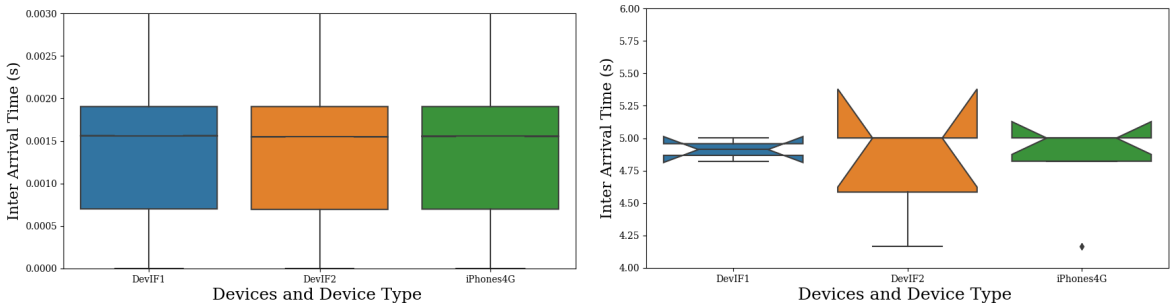
(a) Normal (b) Abnormal

Fig. A.4 The notched box plots of normal and abnormal cluster centroid points for devices (T1-2) and their device type (Asus Tablet) in Active network traffic datasets



(a) Normal (b) Abnormal

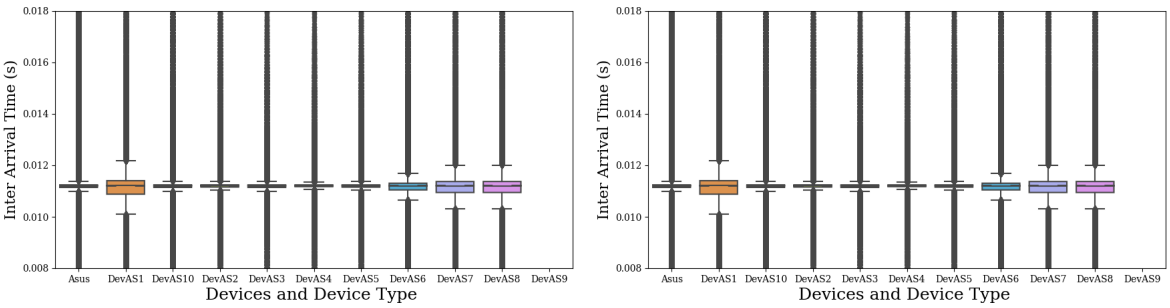
Fig. A.5 The notched box plots of normal and abnormal cluster centroid points for devices (IF1-2) and their device type (iPhone 3G) in Isolated network traffic datasets



(a) Normal (b) Abnormal

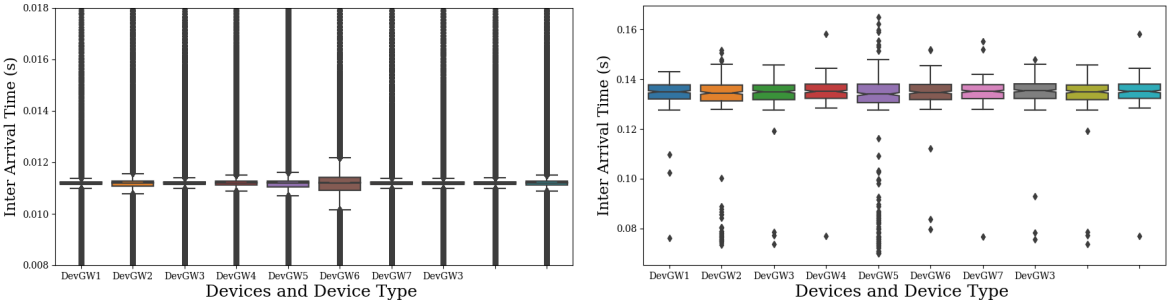
Fig. A.6 The notched box plots of normal and abnormal cluster centroid points for devices (IT1-2) and their device type (iPhone 4G) in Isolated network traffic datasets

Passive Dataset



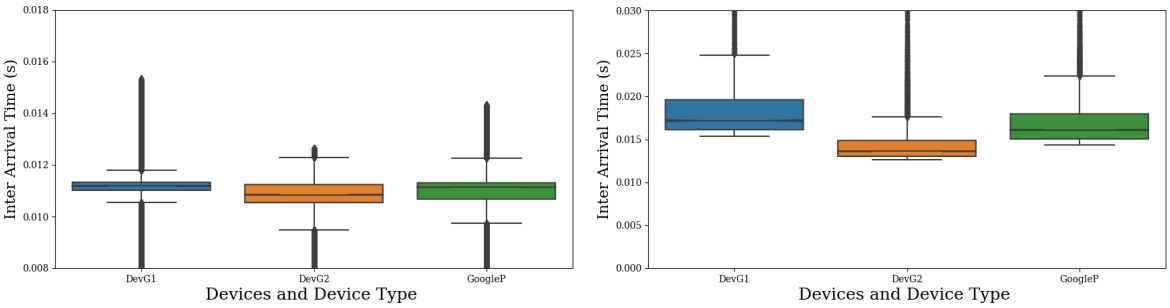
(a) Normal (b) Abnormal

Fig. A.7 The notched box plots of normal and abnormal cluster centroid points for devices (AS1-10) and their device type (Asus) in Passive network traffic dataset



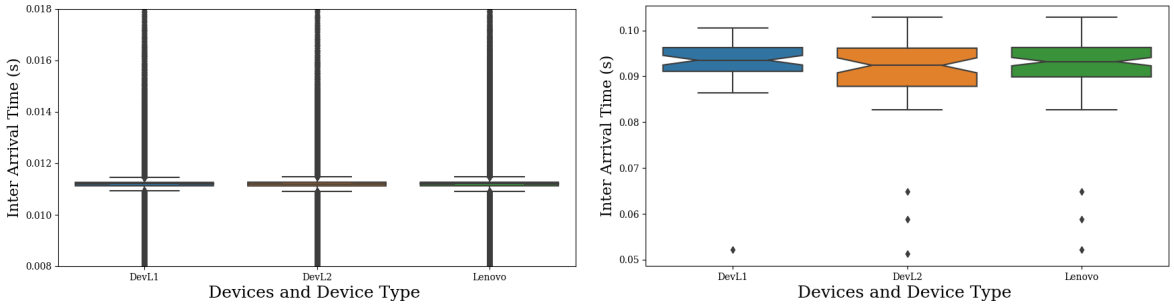
(a) Normal (b) Abnormal

Fig. A.8 The notched box plots of normal and abnormal cluster centroid points for devices (GW1-8) and their device type (Gateway) in Passive network traffic dataset



(a) Normal (b) Abnormal

Fig. A.9 The notched box plots of normal and abnormal cluster centroid points for devices (G1-2) and their device type (Google Phone) in Passive network traffic dataset



(a) Normal (b) Abnormal

Fig. A.10 The notched box plots of normal and abnormal cluster centroid points for devices (L1-2) and their device type (Lenovo) in Passive network traffic dataset

A.3 Device Type Profiling Tables

A.3.1 Device Type Profiles of Active traffic Datasets

Table A.22 A Device Type Profile of Ping-ICMP-Case 1 Active Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer	3,968,592	3,965,611	2,981	0.01	34
Asus	3,969,874	3,967,493	2,381	0.01	32
Gateway NB	3,179,980	3,178,136	1,844	0.01	28
Google Phone	796,817	774,416	22,401	0.3	17
Lenovo	798,309	777,037	21,272	0.3	10
Tablet	794,975	772,017	22,958	0.3	23

Table A.23 A Device Type Profile of Ping-ICMP-Case 2 Active Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer	3,960,087	3,956,597	3,490	0.01	37
Asus	3,966,900	3,964,199	2,701	0.01	36
Gateway NB	3,176,513	3,174,852	1,661	0.01	29
Google Phone	796,565	772,834	23,731	0.3	15
Lenovo	793,563	791,341	2,222	0.01	4
Tablet	794,004	775,337	18,667	0.2	10

A.3.2 Device Type Profiles of Isolated traffic Datasets

Table A.24 A Device Type Profile of iPerf-TCP-Case 2 Isolated Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Dell Netbooks	9,100,324	9,034,201	66,123	0.7	1.35
iPads	4,581,539	4,570,163	11,376	0.2	54
iPhone 3G	1,129,399	1,113,549	15,850	0.1	9
iPhone 4G	8,300,764	8,260,063	40,701	0.5	1.10
Nokia	1,563,011	1,558,888	4,123	0.03	19

Table A.25 A Device Type Profile of iPerf-UDP-Case 1 Isolated Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Dell Netbooks	1,515,860	1,501,041	14,819	0.1	12
iPads	909,939	909,256	6,683	0.07	6
iPhone 3G	628,436	620,255	8,181	0.3	4
iPhone 4G	613,103	612,754	349	0.01	3
Nokia	620,868	610,467	10,401	0.2	4

Table A.26 A Device Type Profile of iPerf-UDP-Case 2 Isolated Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Dell Netbooks	12,110,635	12,030,595	80,040	0.7	1.36
iPads	5,072,177	5,060,025	12,152	0.2	47
iPhone 3G	3,543,093	3,513,112	29,981	0.8	29
iPhone 4G	4,807,669	4,870,140	529	0.001	40
Nokia	3,141,785	3,141,747	38	0.001	24

Table A.27 A Device Type Profile of iPerf-UDP-Case 3 Isolated Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Dell Netbooks	22,111,585	22,024,347	87,238	0.04	4.19
iPads	5,767,239	5,755,094	12,145	0.02	46
iPhone 3G	3,693,552	3,668,007	25,545	0.7	29
iPhone 4G	8,152,218	8,150,471	1,747	0.001	1.06
Nokia	9,375,782	9,319,851	55,931	0.6	1.14

Table A.28 A Device Type Profile of Ping-ICMP-Case 1 Isolated Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Dell Netbooks	1,079,208	1,070,022	9,186	0.09	12
iPads	929,699	922,545	7,154	0.08	12
iPhone 3G	634,786	629,714	5,072	0.08	3
iPhone 4G	1,346,876	1,335,132	11,744	0.09	7
Nokia	718,296	699,750	18,546	0.3	11

Table A.29 A Device Type Profile of Ping-ICMP-Case 2 Isolated Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Dell Netbooks	1,079,984	1,061,970	18,014	0.02	12
iPads	929,699	920,836	7,964	0.01	9
iPhone 3G	634,786	618,618	11,917	0.02	3
iPhone 4G	1,340,598	1,327,588	13,010	0.09	10
Nokia	609,364	592,325	17,039	0.03	8

Table A.30 A Device Type Profile of SCP-TCP-Case 4 Isolated Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Dell Netbooks	7,717,856	7,697,906	18,950	0.3	1.06
iPads	5,247,179	5,237,190	9,989	0.02	45
iPhone 3G	2,197,593	2,180,607	16,906	0.01	23
iPhone 4G	3,198,042	3,197,294	748	0.001	23
Nokia	2,887,527	2,875,990	11,537	0.004	38

A.3.3 Device Type Profiles of Passive traffic Datasets

Table A.31 A Device Type Profile of iPerf-TCP-Case 1 Passive Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer	45,000,000	45,000,000	0	0	6.37
Asus	45,000,000	45,000,000	0	0	6.04
Gateway NB	36,000,000	35,954,206	52,794	0.1	5.11
Google Phone	10,954,547	15,402,952	146,070	1.3	3.40
Lenovo	9,951,385	15,710,481	145,123	1.5	3.36
Tablet	11,636,533	15,097,390	48,428	0.4	5.15

Table A.32 A Device Type Profile of iPerf-UDP-Case 1 Passive Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer	45,000,000	45,000,000	0	0	5.51
Asus	45,000,000	45,000,000	0	0	5.20
Gateway NB	36,000,000	36,000,000	0	0	3.35
Google Phone	15,598,782	15,402,952	195,830	1.3	3.25
Lenovo	15,954,580	15,710,481	244,099	1.5	1.57
Tablet	15,340,910	15,097,390	243,520	1.6	4.53

Table A.33 A Device Type Profile of iPerf-UDP-Case 2 Passive Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Lenovo	24,555,234	24,140,905	414,329	1.7	2.37
Tablet	12,635,530	12,571,571	63,959	0.5	1.16

Table A.34 A Device Type Profile of iPerf-UDP-Case 3 Passive Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer	3,212,437	3,193,800	18,637	0.06	15
Asus	3,212,202	3,202,305	9,867	0.03	14
Gateway NB	2,569,754	2,566,525	13,229	0.05	12
Google Phone	650,215	634,293	15,922	0.24	3
Lenovo	642,587	636,035	6,552	0.10	3
Tablet	642,502	625,439	17,063	0.27	10

Table A.35 A Device Type Profile of Ping-ICMP-Case 1 Passive Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer	3,946,612	3,927,516	19,096	0.5	23
Asus	3,952,631	3,949,211	3,420	0.1	21
Gateway NB	3,159,515	3,142,926	16,589	0.5	19
Google Phone	483,137	479,484	3,653	0.8	3
Lenovo	799,001	791,056	7,945	1	4
Tablet	703,215	681,710	21,505	3.1	11

Table A.36 A Device Type Profile of Ping-ICMP-Case 2 Passive Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer	3,935,292	3,913,797	21,499	0.5	23
Asus	3,936,855	3,921,189	15,666	0.4	23
Gateway NB	3,150,608	3,136,357	14,251	0.5	19
Google Phone	560,063	556,851	3,212	0.6	3
Lenovo	797,296	786,356	10,940	1.4	4
Tablet	694,979	675,791	19,188	2.8	9

Table A.37 A Device Type Profile of SCP-TCP-Case 1 Passive Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer	45,000,000	44,992,290	7,710	0.01	5.47
Asus	45,000,000	45,000,000	0	0	5.51
Gateway NB	36,000,000	35,971,240	28,760	0.1	4.18
Google Phone	2,761,229	2,732,776	28,453	1	18
Lenovo	8,816,319	8,658,176	158,143	1.8	3.09
Tablet	7,177,578	6,915,337	262,241	3.7	55

Table A.38 A Device Type Profile of Skype-UDP-Case 1 Passive Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Lenovo	1,102,415	1,057,680	44,735	4.1	1
Tablet	1,014,217	1,010,920	3,297	0.3	6

A.3.4 Device Type profile Plots for Isolated Network Traffic Dataset

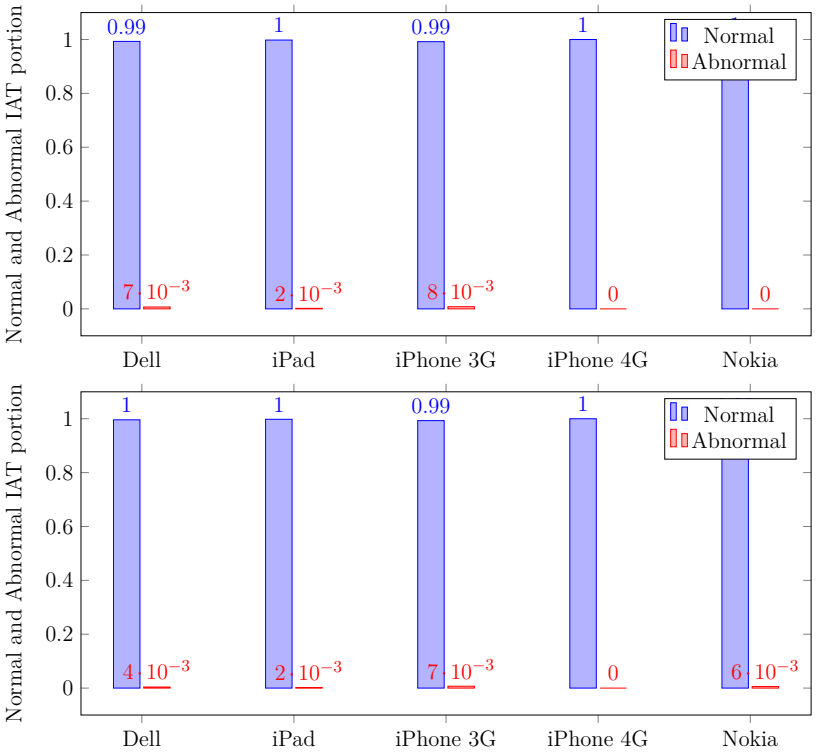


Fig. A.11 Normal and Abnormal Device Type Profiles of iPerf-UDP-case 2 and 3 Datasets

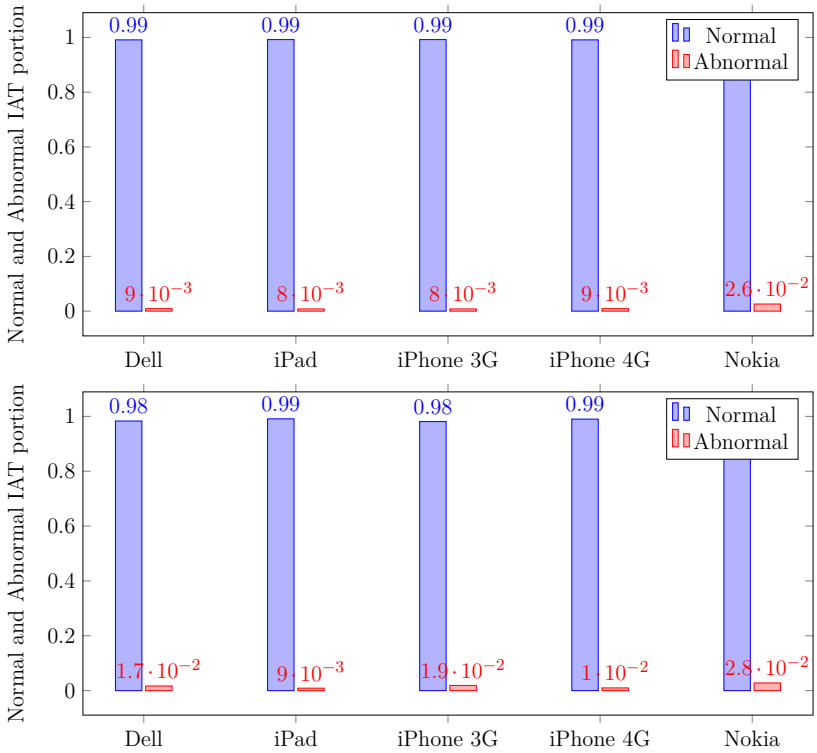


Fig. A.12 Normal and Abnormal Device Type Profiles of Ping-ICMP-case 1 and 2 Datasets

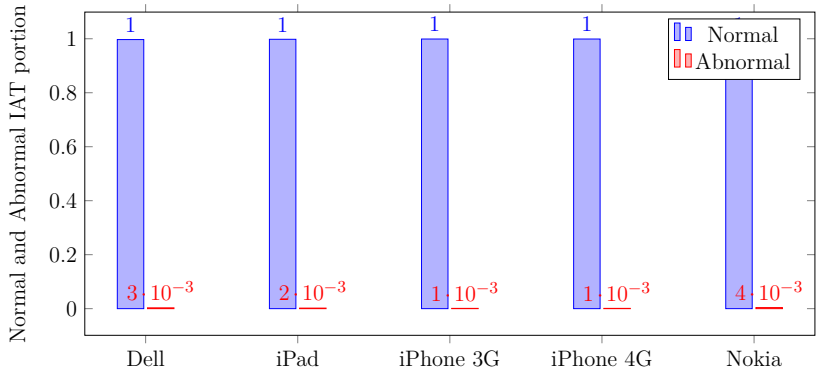


Fig. A.13 Normal and Abnormal Device Type Profiles of SCP-TCP-Case 4 Dataset

A.3.5 Device Type profile Plots for Passive Network Traffic Dataset

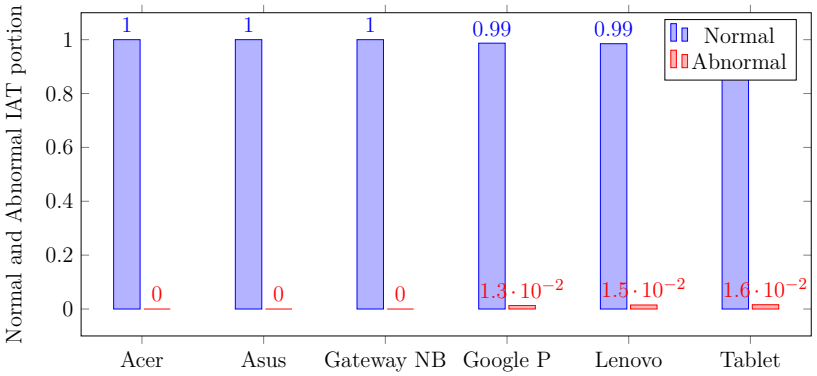


Fig. A.14 Normal and Abnormal Device Type Profiles of iPerf UDP Case 1 Datasets

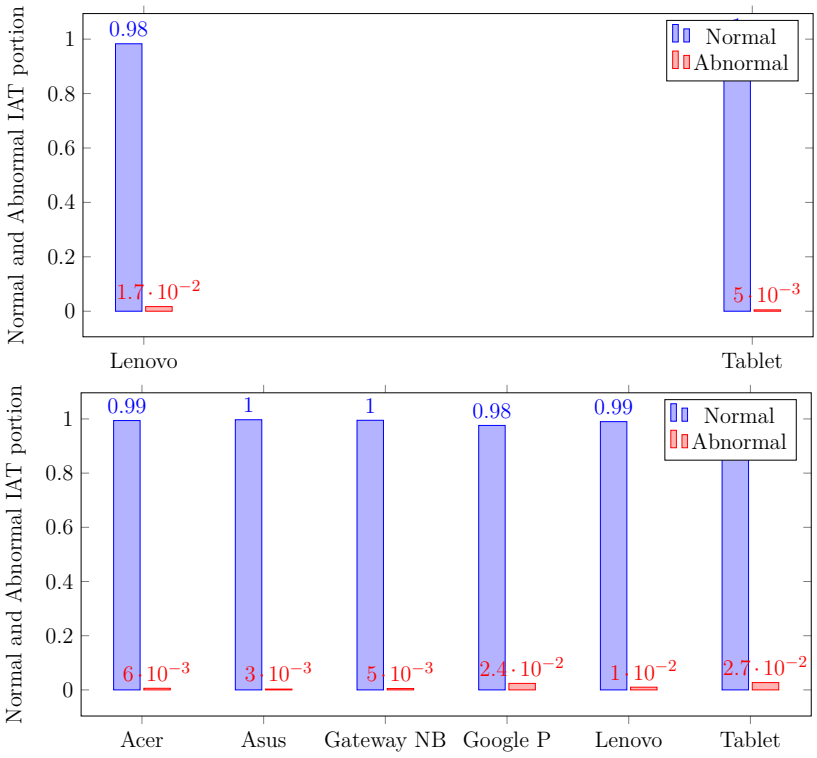


Fig. A.15 Normal and Abnormal Device Type Profiles of iPerf-UDP-Case 2 and 3 Datasets

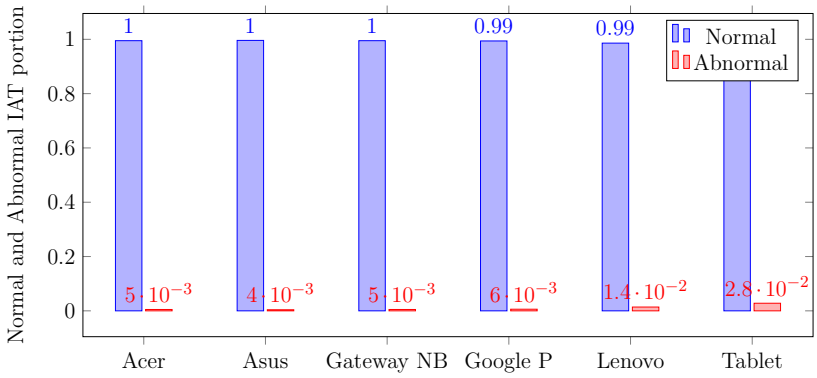


Fig. A.16 Normal and Abnormal Device Type Profiles of Ping-ICMP-Case 2 Dataset

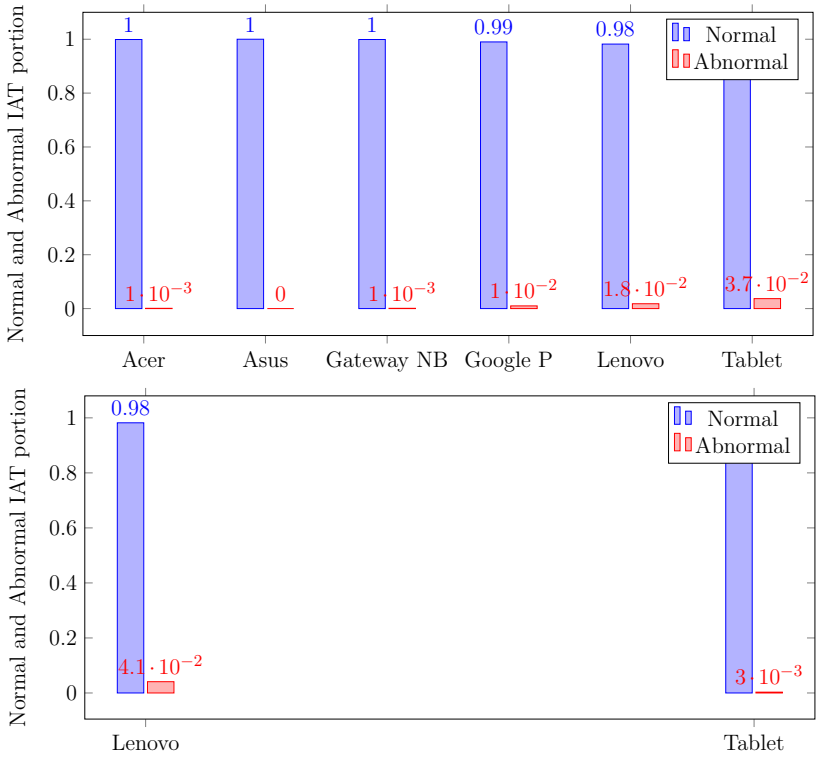


Fig. A.17 Normal and Abnormal Device Type Profiles of SCP-TCP Case 1 and Skype-UDP-Case 1 Datasets

Appendix B

Intelligent Filtering Technique

B.1 Intelligent Filtering Technique Results

B.1.1 Active Network Traffic

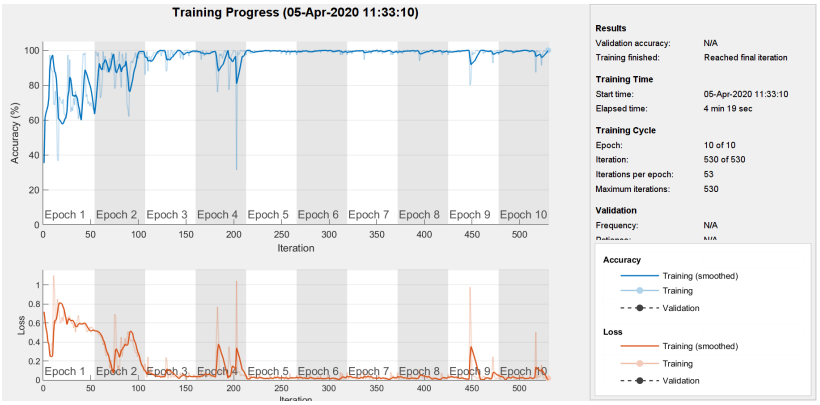


Fig. B.1 The intelligent filtering technique training progress for Acer Netbook

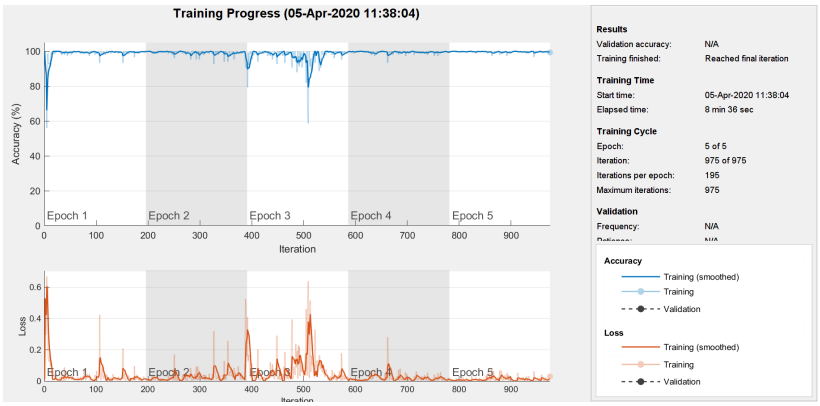


Fig. B.2 The intelligent filtering technique additional training progress for Acer Netbook

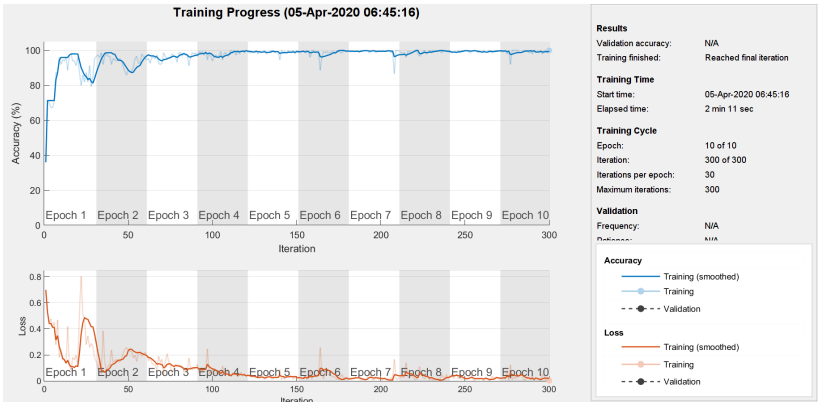


Fig. B.3 The intelligent filtering technique training progress for Asus Netbook

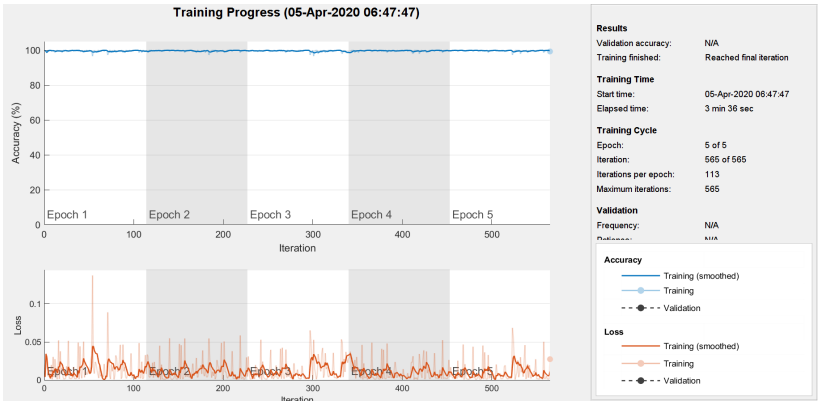


Fig. B.4 The intelligent filtering technique additional training progress for Asus Netbook

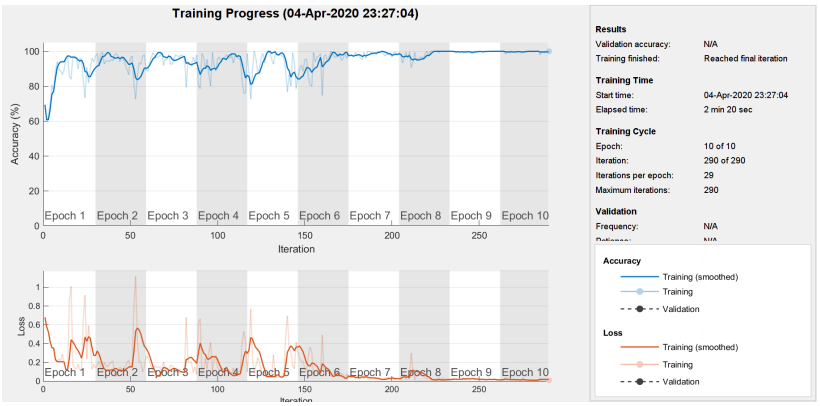


Fig. B.5 The intelligent filtering technique training progress for Gateway Netbook

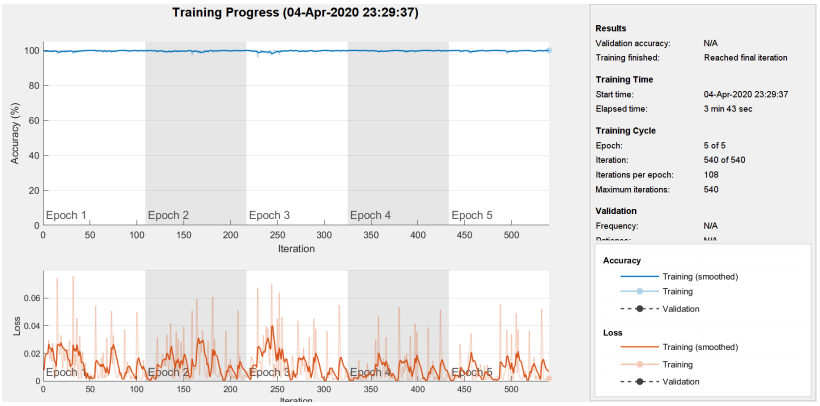


Fig. B.6 The intelligent filtering technique additional training progress for Gateway Netbook

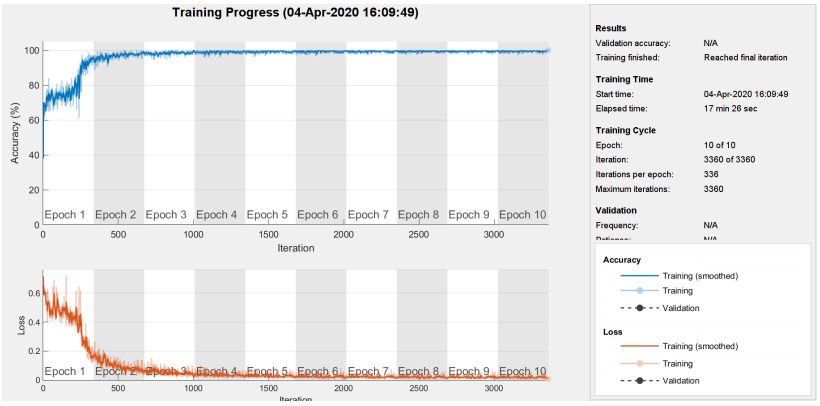


Fig. B.7 The intelligent filtering technique training progress for Google Phone

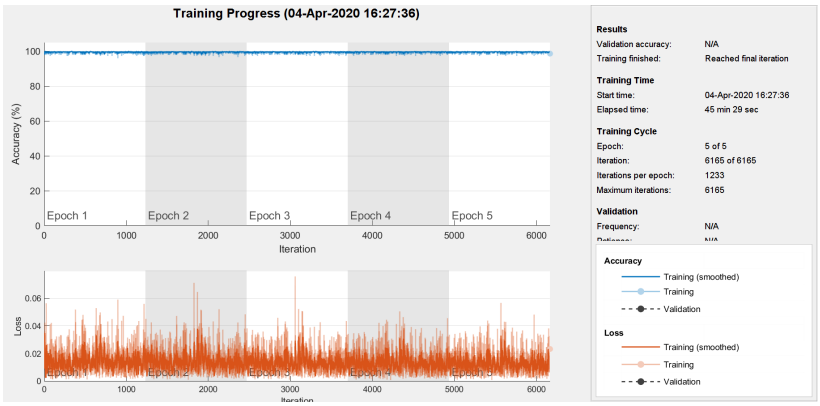


Fig. B.8 The intelligent filtering technique additional training progress for Google Phone

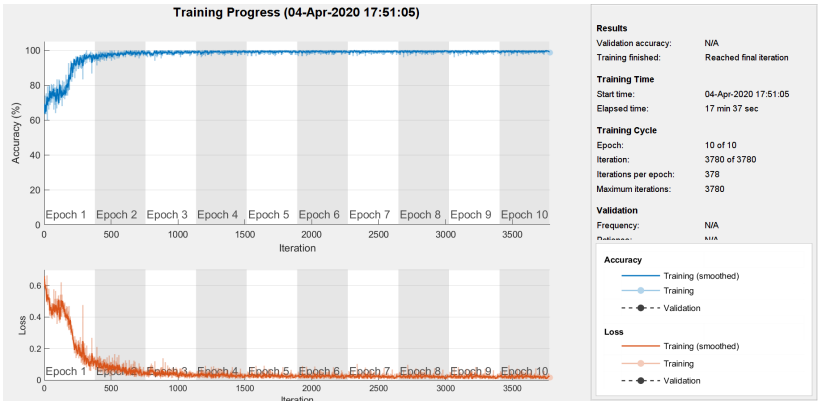


Fig. B.9 The intelligent filtering technique training progress for Lenovo Laptop

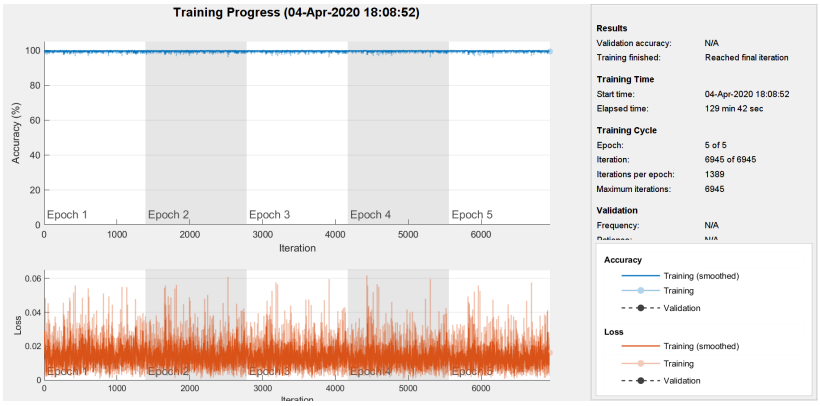


Fig. B.10 The intelligent filtering technique additional training progress for Lenovo Laptop

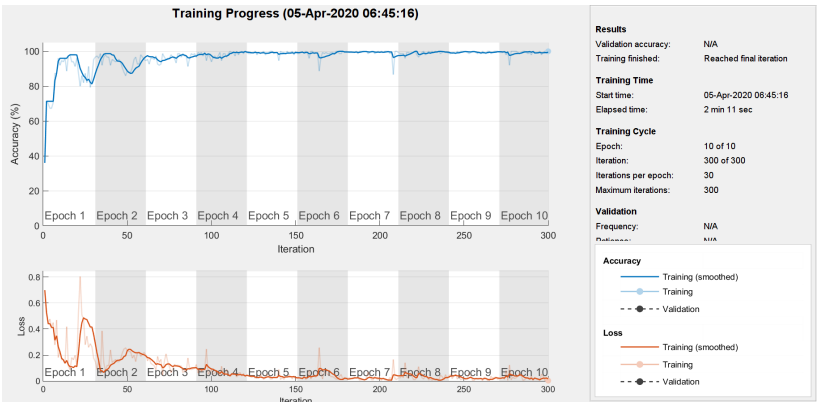


Fig. B.11 The intelligent filtering technique training progress for Asus Asus Tablet

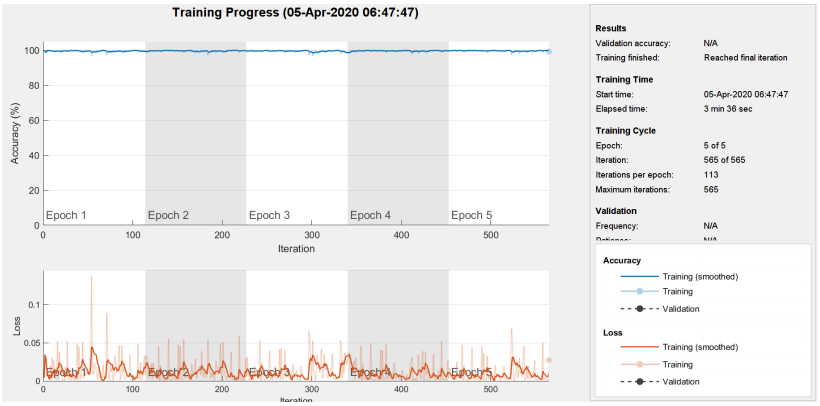


Fig. B.12 The intelligent filtering technique additional training progress for Asus Asus Tablet

B.1.2 isolated Network Traffic

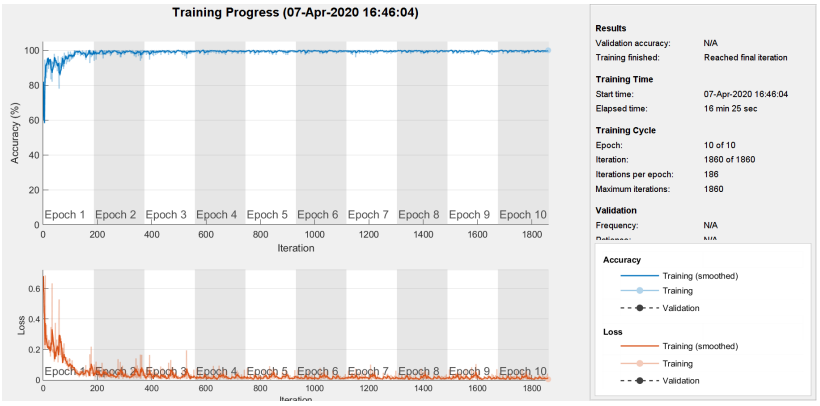


Fig. B.13 The intelligent filtering technique training progress for iPad

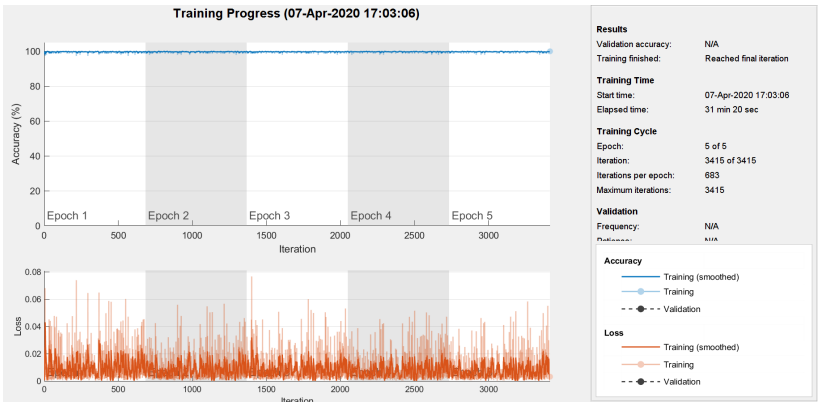


Fig. B.14 The intelligent filtering technique additional training progress for iPad

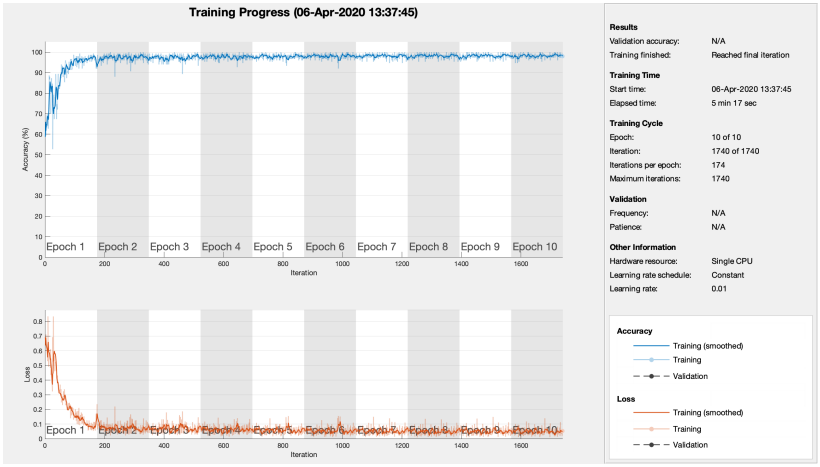


Fig. B.15 The intelligent filtering technique training progress for iPhone 3G

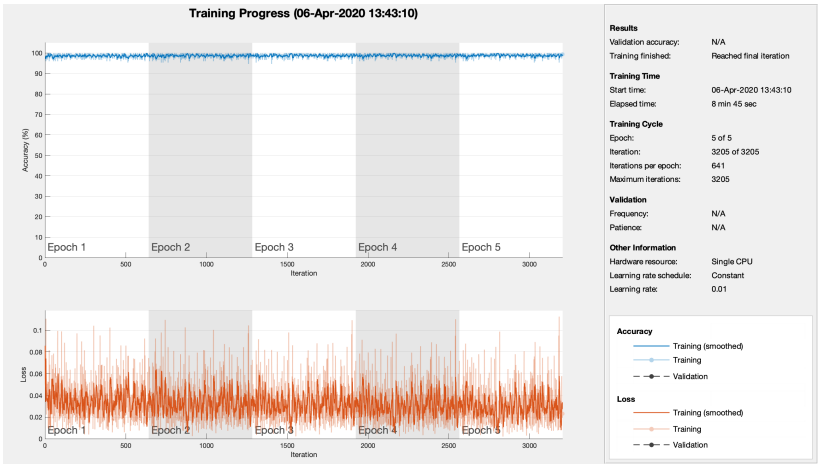


Fig. B.16 The intelligent filtering technique additional training progress for iPhone 3G

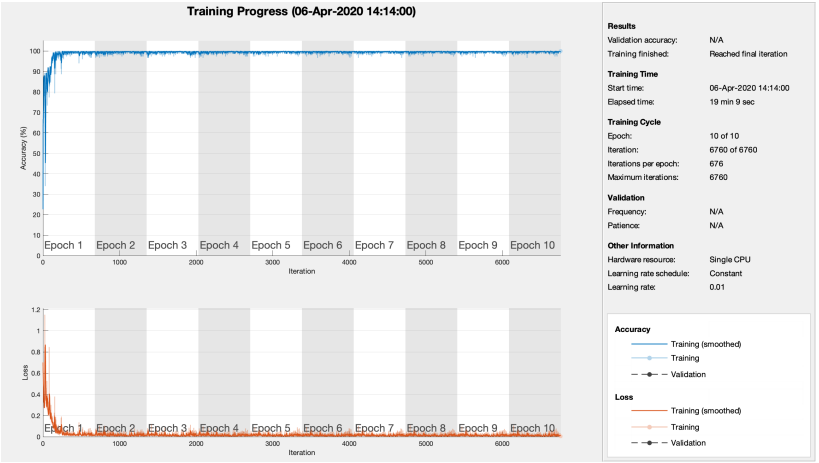


Fig. B.17 The intelligent filtering technique training progress for iPhone 4G

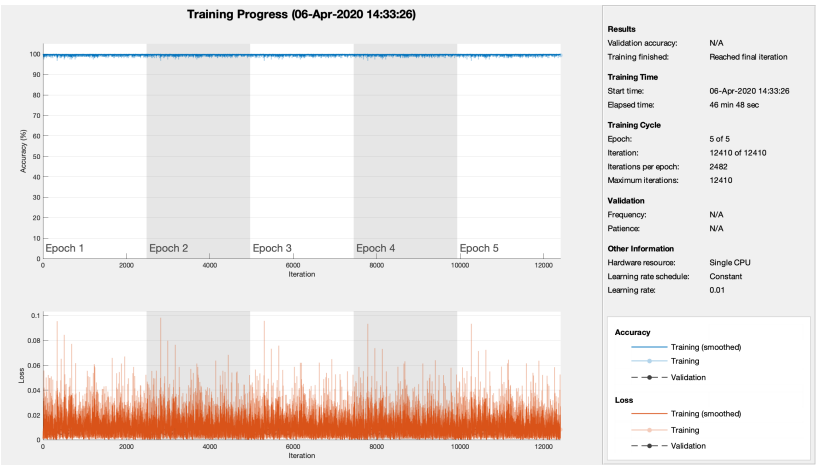


Fig. B.18 The intelligent filtering technique additional training progress for iPhone 4G

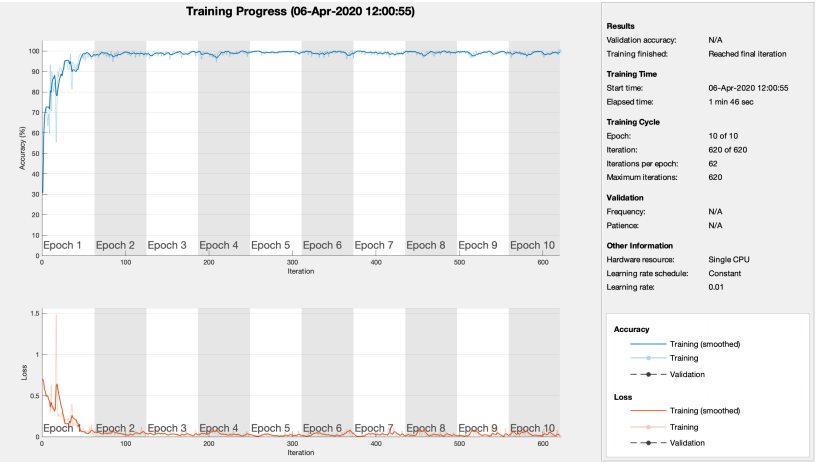


Fig. B.19 The intelligent filtering technique training progress for Nokia Phone

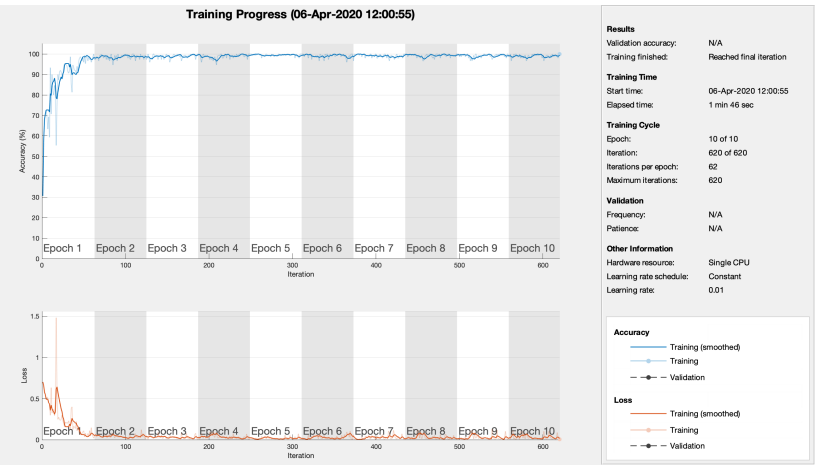


Fig. B.20 The intelligent filtering technique additional training progress for Nokia Phone

B.1.3 Passive Network Traffic

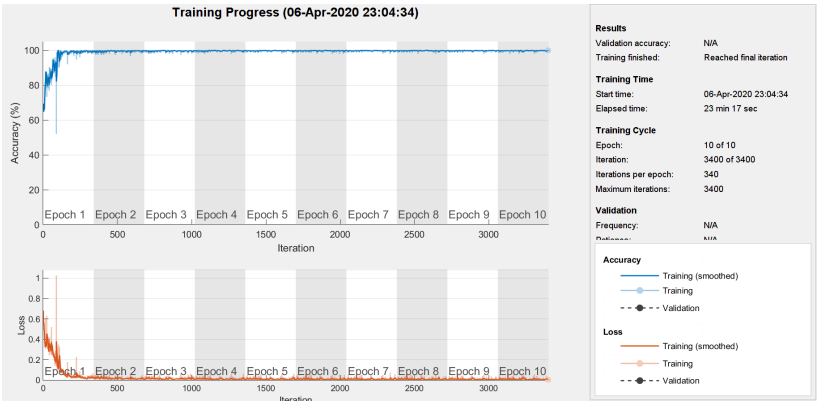


Fig. B.21 The intelligent filtering technique training progress for Acer Netbook

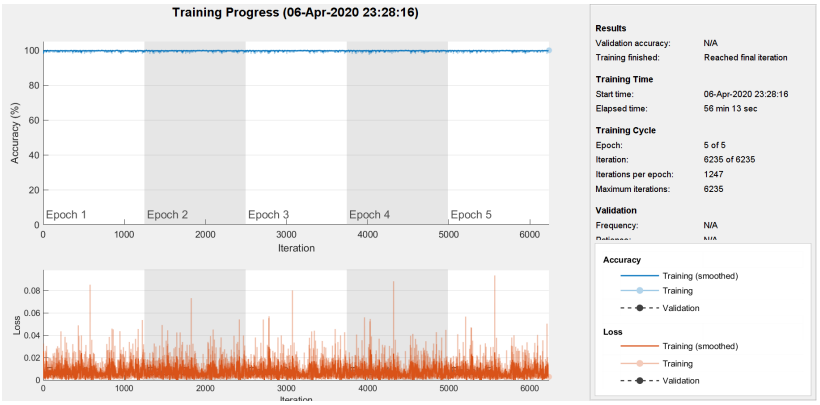


Fig. B.22 The intelligent filtering technique additional training progress for Acer Netbook

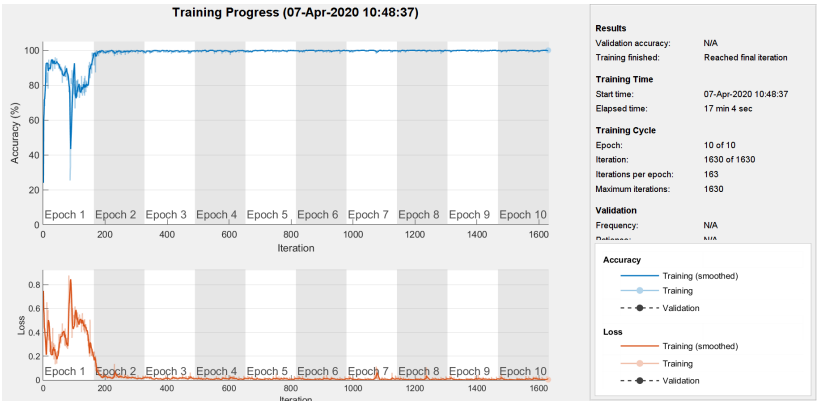


Fig. B.23 The intelligent filtering technique training progress for Asus Netbook

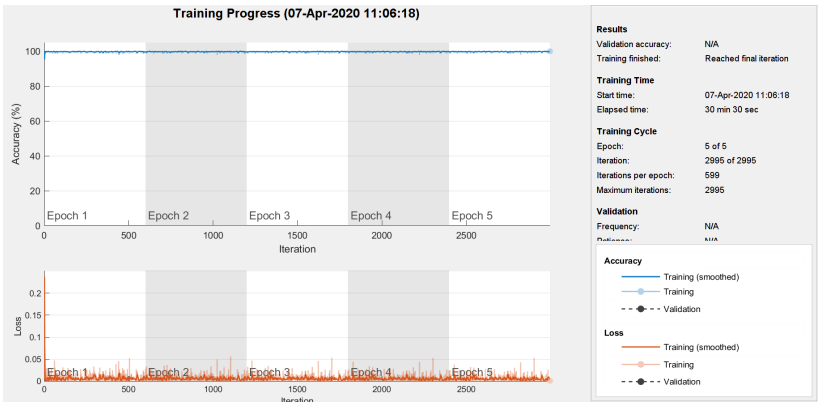


Fig. B.24 The intelligent filtering technique additional training progress for Asus Netbook

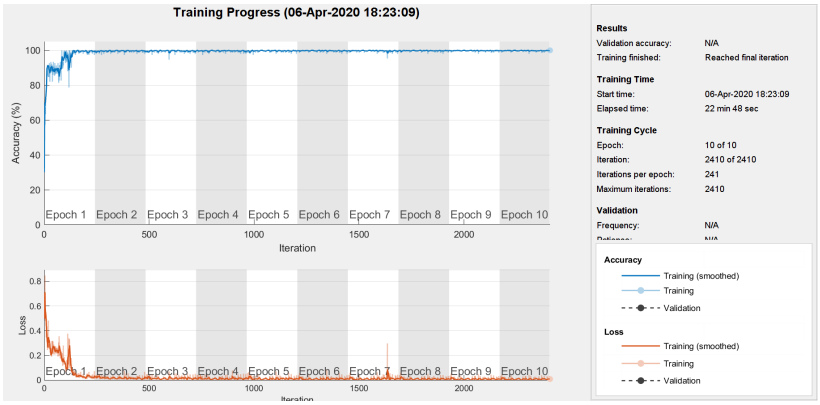


Fig. B.25 The intelligent filtering technique training progress for Gateway Netbook

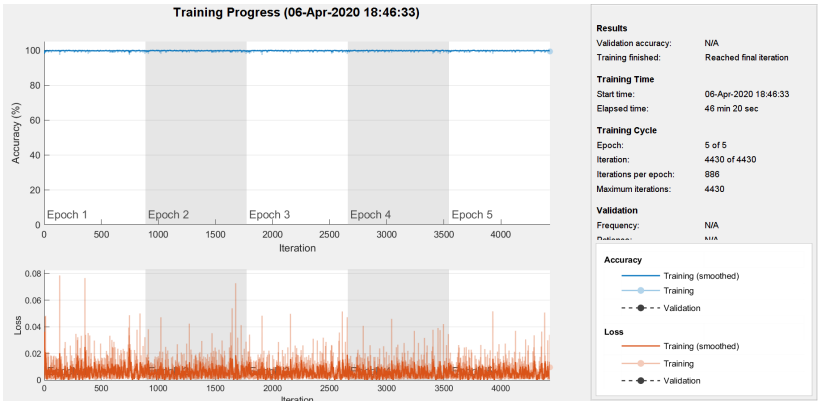


Fig. B.26 The intelligent filtering technique additional training progress for Gateway Netbook

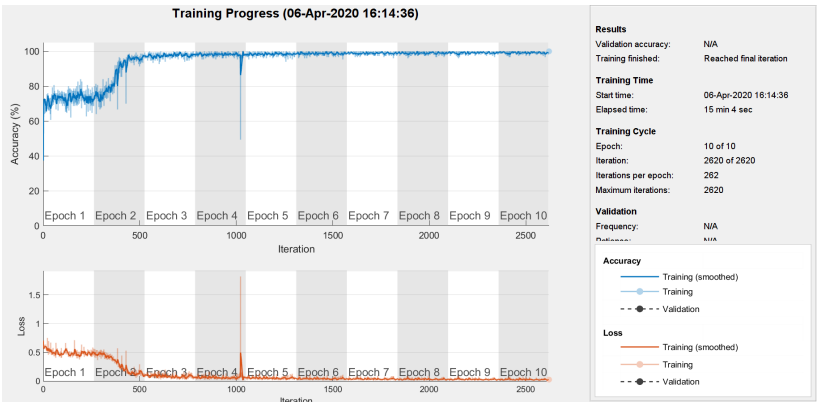


Fig. B.27 The intelligent filtering technique training progress for Google Phone

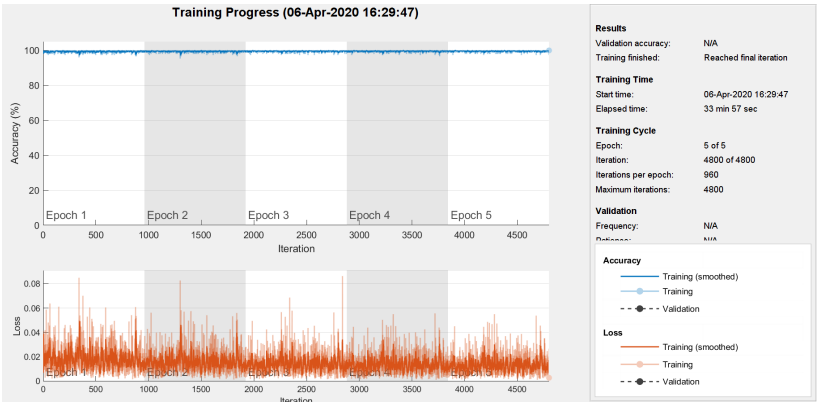


Fig. B.28 The intelligent filtering technique additional training progress for Google Phone

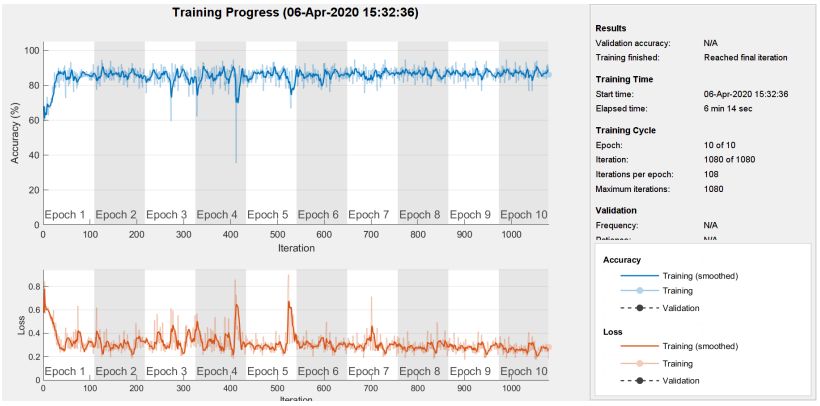


Fig. B.29 The intelligent filtering technique training progress for Lenovo Laptop

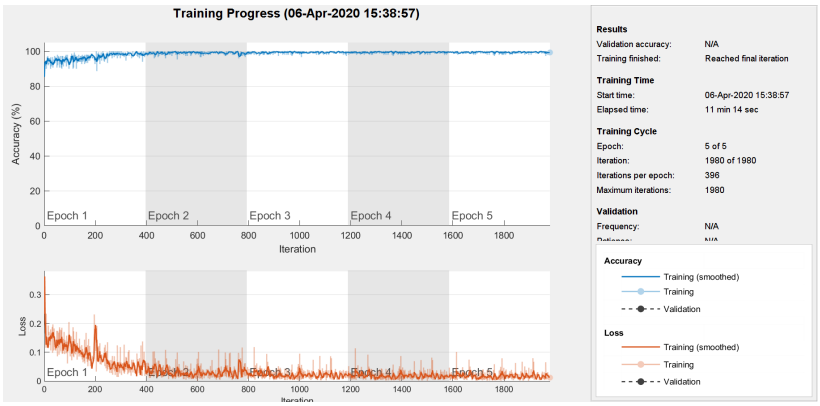


Fig. B.30 The intelligent filtering technique additional training progress for Lenovo Laptop

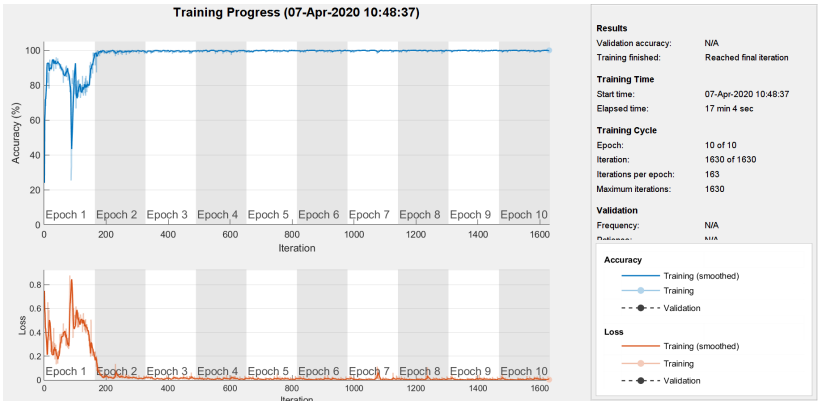


Fig. B.31 The intelligent filtering technique training progress for Asus Tablet

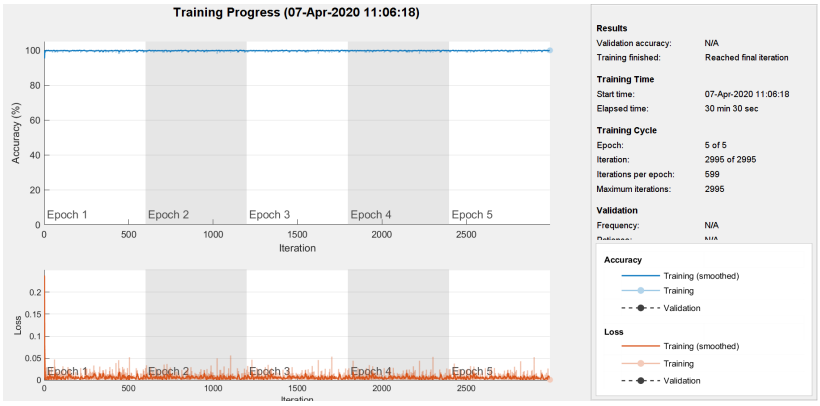


Fig. B.32 The intelligent filtering technique additional training progress for Asus Tablet

B.2 Synthetic Data Generation

B.2.1 Probability Distribution Fitting

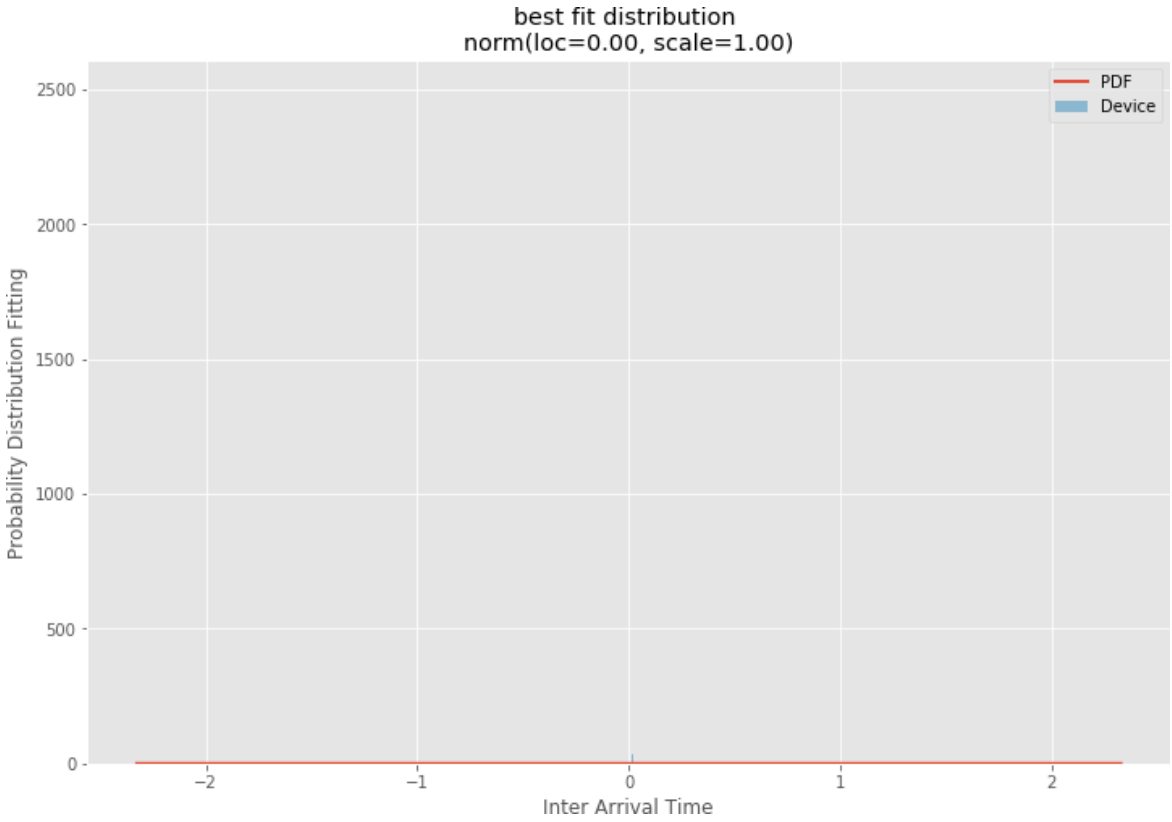


Fig. B.33 The sample probability distribution fitting for iPhone 4G

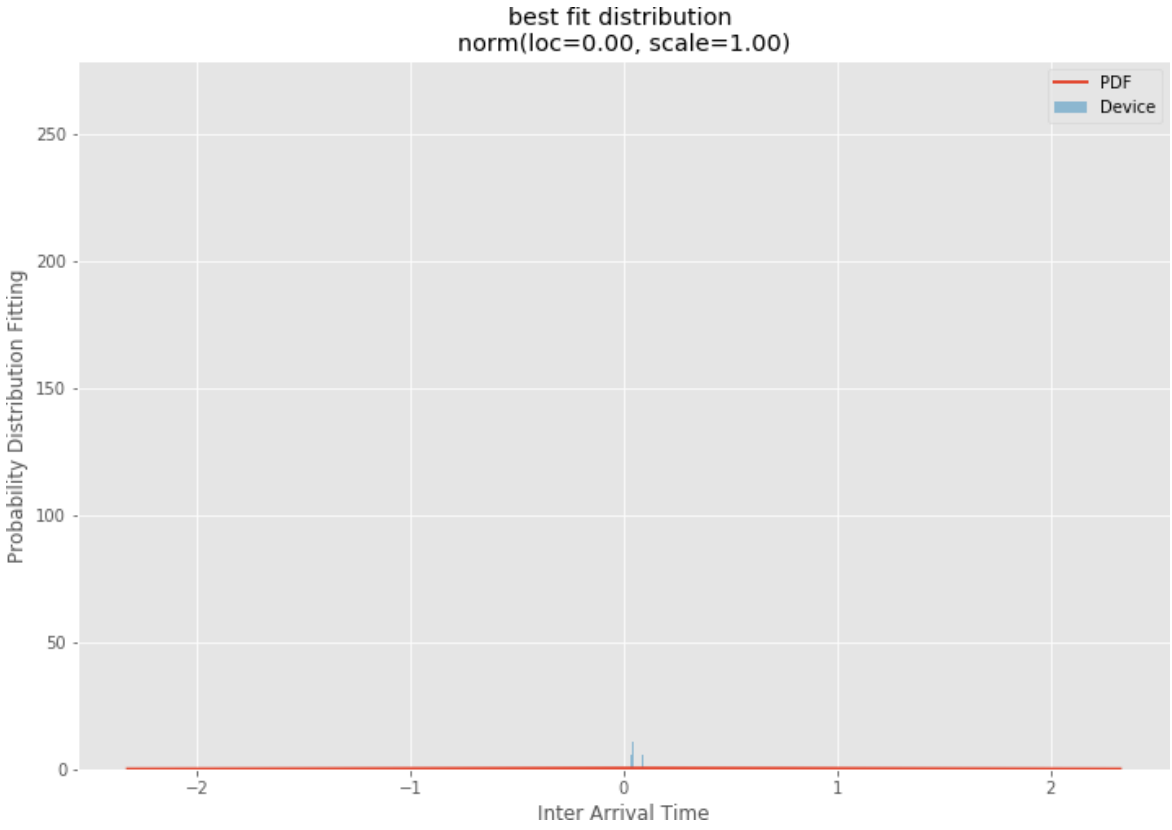


Fig. B.34 The sample probability distribution fitting for iPhone 3G

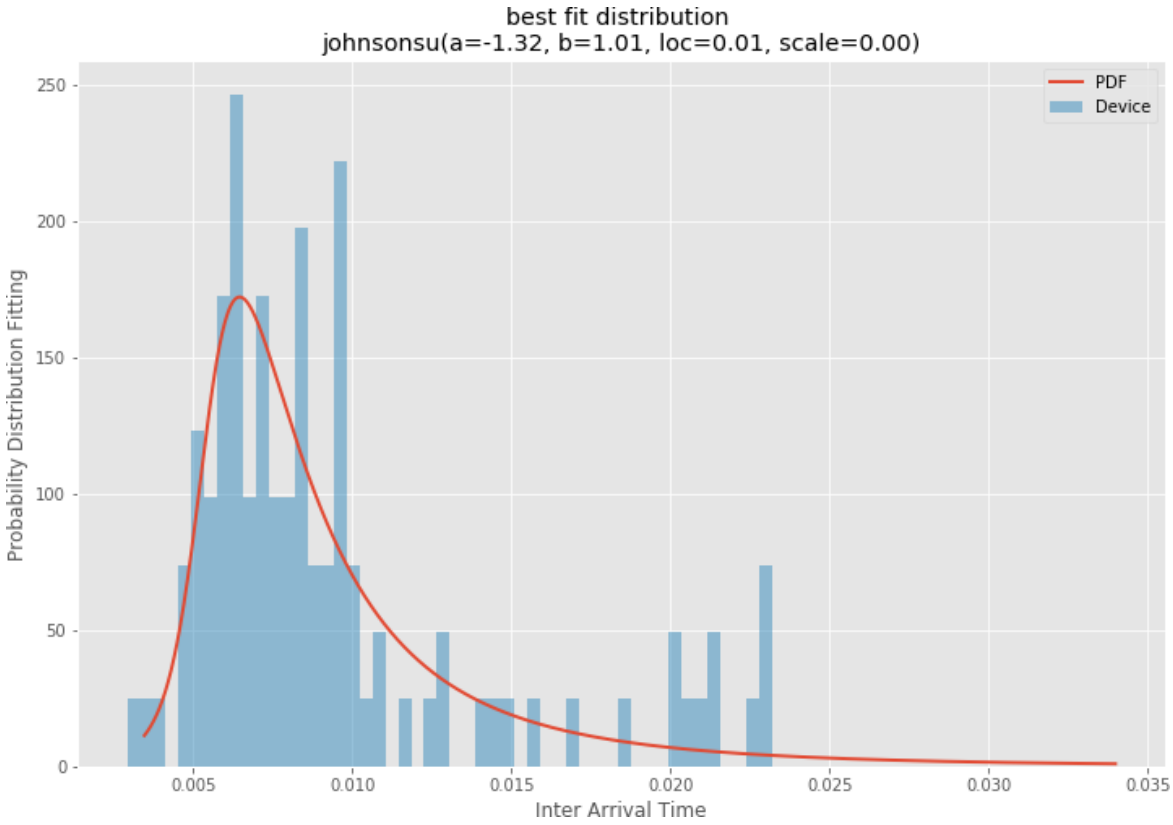


Fig. B.35 The sample best fitted distribution for Dell Netbook (DN5)

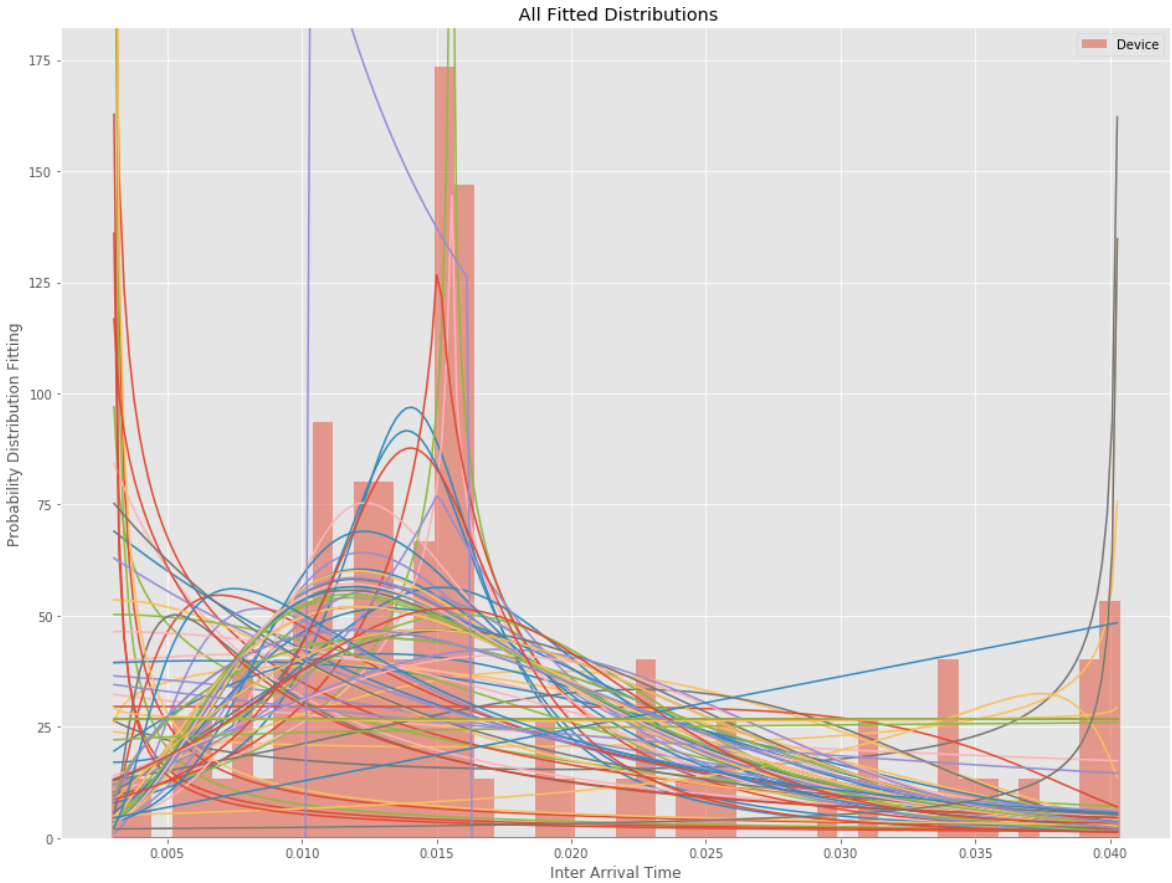


Fig. B.36 The sample best fitted distribution for Dell Netbook (DN4)

B.2.2 Curve Fitting

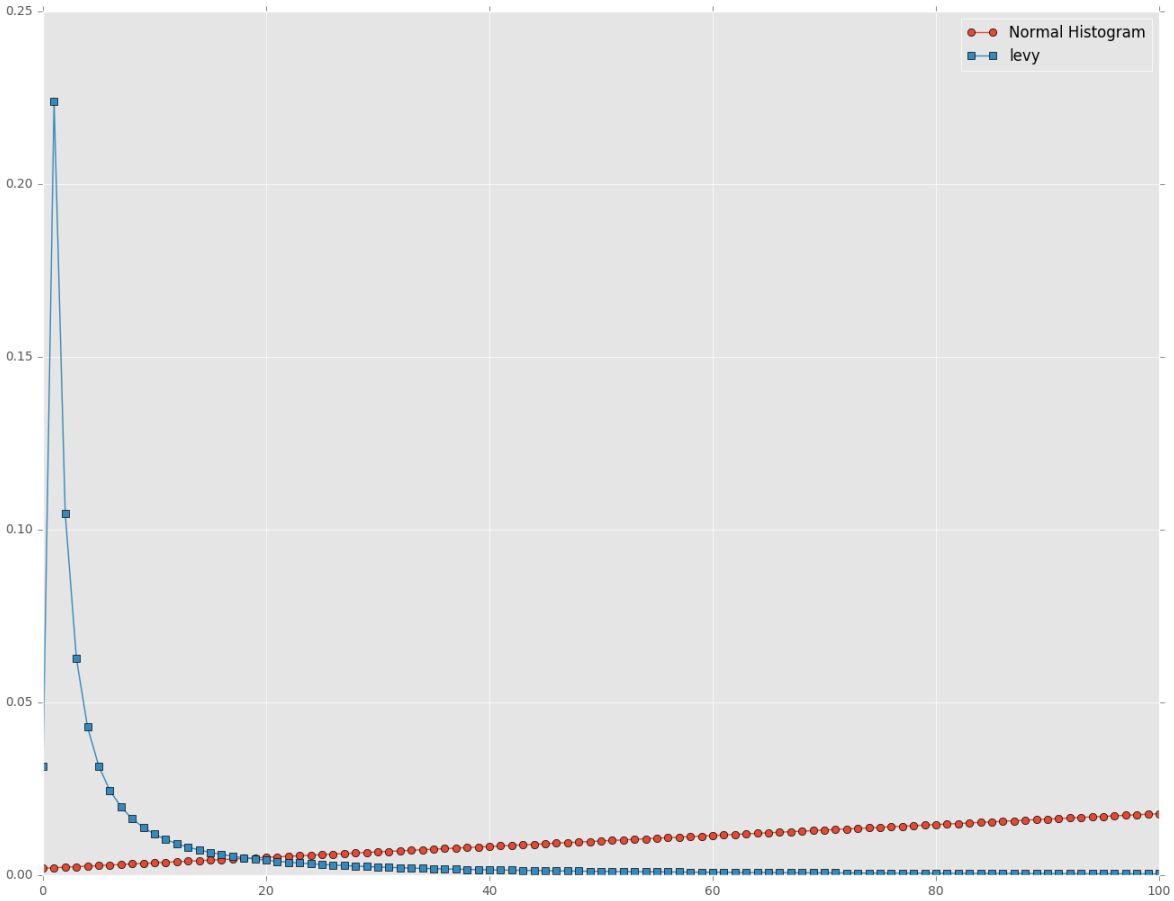


Fig. B.37 The Sample Curve Fitting for Dell Netbooks

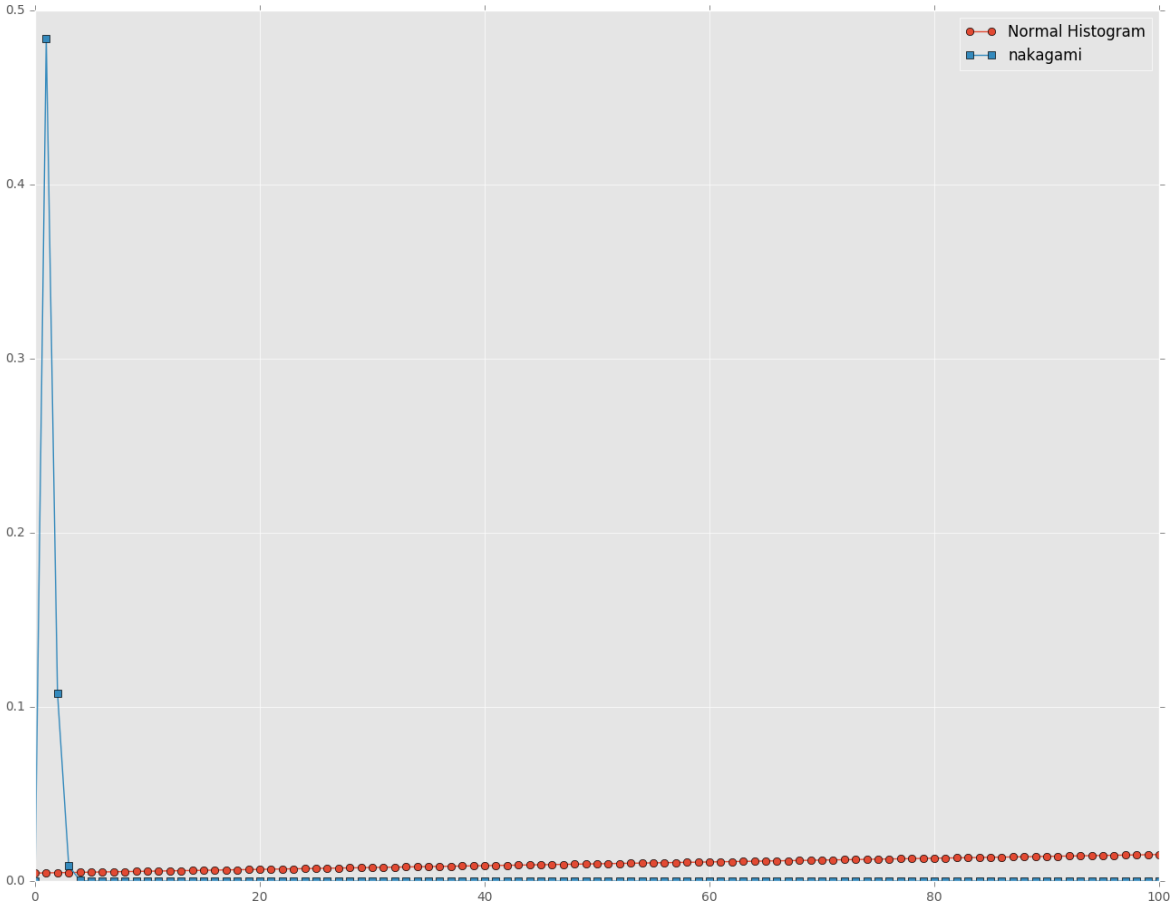


Fig. B.38 The Sample Curve Fitting for iPad

B.3 Synthetic Data Analysis

B.3.1 Active Traffic Dataset

Table B.1 A Device Type Profile of Ping-ICMP-Case 1 Active Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer Netbook	3,968,592	3,967,712	880	0.000	18
Asus Netbook	3,969,874	3,968,941	933	0.000	15
Gateway NB	3,179,980	3,177,776	2,204	0.001	13
Google Phone	796,817	775,799	21,018	0.026	12
Lenovo Laptop	798,309	777,532	20,777	0.026	3
Asus Tablet	794,975	775,918	19,057	0.024	3

Table B.2 A Device Type Profile of Ping-ICMP-Case 2 Active Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer Netbook	3,960,087	3,956,798	3,379	0.001	17
Asus Netbook	3,966,900	3,965,662	1,238	0.001	17
Gateway NB	3,176,513	3,174,421	2,092	0.001	15
Google Phone	796,565	773,631	22,934	0.029	16
Lenovo Laptop	793,563	792,992	571	0.001	3
Asus Tablet	794,004	777,797	16,207	0.020	4

B.3.2 Isolated

Table B.3 A Device Type Profile of iPerf-TCP-Case 2 Isolated Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Dell Netbooks	9,100,324	9,079,941	20,383	0.2	56
iPads	4,581,539	4,572,896	8,643	0.2	28
iPhone 3G	1,129,399	1,114,005	15,394	0.4	5
iPhone 4G	8,300,764	8,258,235	42,529	0.5	41
Nokia	1,563,011	1,562,535	476	0.001	9

Table B.4 A Device Type Profile of iPerf-UDP-Case 1 Isolated Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Dell Netbooks	1,515,860	1,503,365	12,495	0.8	16
iPads	909,939	904,018	5,921	0.7	3
iPhone 3G	628,436	620,184	8,252	0.1	2
iPhone 4G	613,103	612,363	340	0.01	2
Nokia	620,868	604,059	16,809	0.3	2

Table B.5 A Device Type Profile of iPerf-UDP-Case 2 Isolated Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Dell Netbooks	12,110,635	12,105,723	4,912	0.001	50
iPads	5,072,177	5,064,519	7,568	0.2	21
iPhone 3G	3,543,093	3,520,521	22,572	0.6	16
iPhone 4G	4,807,669	4,870,356	313	0.001	19
Nokia	3,141,785	3,141,733	52	0.001	11

Table B.6 A Device Type Profile of iPerf-UDP-Case 3 Isolated Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Dell Netbooks	22,111,585	22,094,304	2,281	0.001	1.41
iPads	5,767,239	5,759,559	7,680	0.01	24
iPhone 3G	3,693,552	3,680,397	13,155	0.4	15
iPhone 4G	8,152,218	8,151,849	369	0.001	36
Nokia	9,375,782	9,375,603	179	0.001	41

Table B.7 A Device Type Profile of Ping-ICMP-Case 1 Isolated Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Dell Netbooks	1,079,208	1,069,850	9,358	0.09	7
iPads	929,699	922,389	7,310	0.08	8
iPhone 3G	634,786	629,737	5,049	0.08	2
iPhone 4G	1,346,876	1,342,775	4,101	0.09	5
Nokia	718,296	700,178	18,118	0.3	9

Table B.8 A Device Type Profile of Ping-ICMP-Case 2 Isolated Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Dell Netbooks	1,079,984	1,060,982	19,002	0.2	5
iPads	929,699	921,062	7,738	0.08	6
iPhone 3G	634,786	620,099	10,436	0.2	2
iPhone 4G	1,340,598	1,328,980	11,618	0.9	5
Nokia	609,364	592,571	16,793	0.3	4

Table B.9 A Device Type Profile of SCP-TCP-Case 4 Isolated Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Dell Netbooks	7,717,856	7,717,731	125	0.001	43
iPads	5,247,179	5,242,413	4,766	0.001	21
iPhone 3G	2,197,593	2,184,460	13,053	0.06	9
iPhone 4G	3,198,042	3,197,620	422	0.001	13
Nokia	2,887,527	2,877,004	10,523	0.04	22

B.3.3 Passive Dataset

Table B.10 A Device Type Profile of iPerf-TCP-Case 1 Passive Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer Netbook	45,000,000	44,998,502	1,498	0.001	6.11
Asus Netbook	45,000,000	44,998,503	1,497	0.001	5.50
Gateway NB	36,000,000	35,998,444	1,556	0.001	3.58
Google Phone	10,954,547	10,801,581	152,966	1.4	3.19
Lenovo Laptop	9,951,385	9,831,216	120,196	1.2	3.26
Asus Tablet	11,636,533	11,479,367	157,166	1.4	3.20

Table B.11 A Device Type Profile of iPerf-UDP-Case 1 Passive Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer Netbook	45,000,000	44,804,507	195,493	0.4	4.56
Asus Netbook	45,000,000	44,943,410	56,490	0.01	4.43
Gateway NB	36,000,000	35,993,794	6,206	0.001	3.03
Google Phone	15,598,782	15,503,278	95,504	0.6	2.29
Lenovo Laptop	15,954,580	15,761,201	193,379	1.2	1.20
Asus Tablet	15,340,910	15,104,289	236,621	1.5	4.10

Table B.12 A Device Type Profile of iPerf-UDP-Case 2 Passive Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Lenovo Laptop	24,555,234	24,544,550	10,684	0.001	2.49
Asus Tablet	12,635,530	12,634,492	5,038	0.001	1.00

Table B.13 A Device Type Profile of iPerf-UDP-Case 3 Passive Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer Netbook	3,212,437	3,193,708	18,729	0.6	16
Asus Netbook	3,212,202	3,202,876	9,326	0.03	16
Gateway NB	2,569,754	2,556,205	13,549	0.05	11
Google Phone	650,215	636,922	13,293	0.02	3
Lenovo Laptop	642,587	636,011	6,576	0.01	2
Asus Tablet	642,502	626,036	16,466	0.3	12

Table B.14 A Device Type Profile of Ping-ICMP-Case 1 Passive Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer Netbook	3,946,612	3,926,621	19,991	0.05	17
Asus Netbook	3,952,631	3,952,631	0	0	50
Gateway NB	3,159,515	3,144,295	15,220	0.05	15
Google Phone	483,137	479,115	4,022	0.08	2
Lenovo Laptop	799,001	791,336	7,665	0.01	3
Asus Tablet	703,215	682,681	20,534	0.03	13

Table B.15 A Device Type Profile of Ping-ICMP-Case 2 Passive Traffic Dataset

Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer	3,935,292	3,914,252	21,040	0.05	17
Asus Netbook	3,936,855	3,922,436	14,419	0.04	18
Gateway NB	3,150,608	3,135,235	15,373	0.05	13
Google Phone	560,063	556,627	3,436	0.006	2
Lenovo Laptop	797,296	786,658	10,638	0.001	3
Asus Tablet	694,979	678,709	16,270	0.002	8

Table B.16 A Device Type Profile of SCP-TCP-Case 1 Passive Traffic Dataset

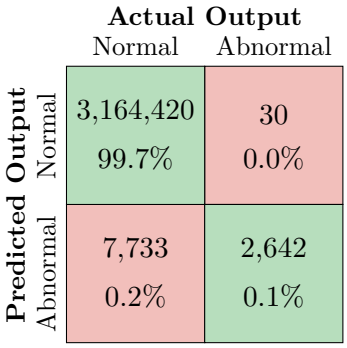
Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Acer Netbook	45,000,000	44,999,659	341	0.001	4.15
Asus Netbook	45,000,000	44,998,461	1,539	0.001	7.38
Gateway NB	36,000,000	35,997,731	2,269	0.001	4.12
Google Phone	2,761,229	2,734,066	27,163	0.01	17
Lenovo	8,816,319	8,671,336	144,983	1.6	1.20
Asus Tablet	7,177,578	6,944,623	232,955	3.2	48

Table B.17 A Device Type Profile of Skype-UDP-Case 1 Passive Traffic Dataset

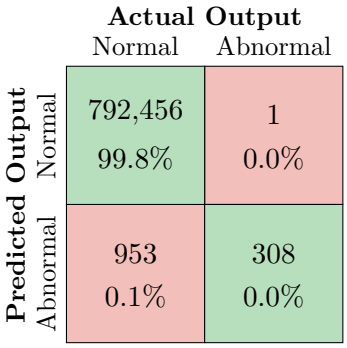
Device Type	IAT points	Normal points	Abnormal points	% abnormal	Run Time (s)
Lenovo Laptop	1,102,415	1,057,214	45,201	0.4	13
Asus Tablet	1,014,217	1,004,789	9,428	0.009	7

B.4 Intelligent Filtering Synthetic Data Evaluation

B.4.1 Evaluation results for Active Network traffic Datasets

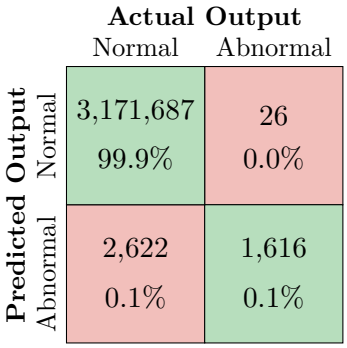


(a) Training Confusion Matrix

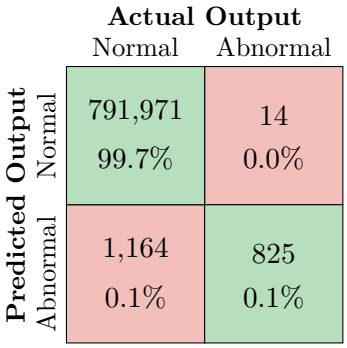


(b) Testing Confusion Matrix

Fig. B.39 The intelligent filtering technique Evaluation Confusion Matrix for Acer Netbook

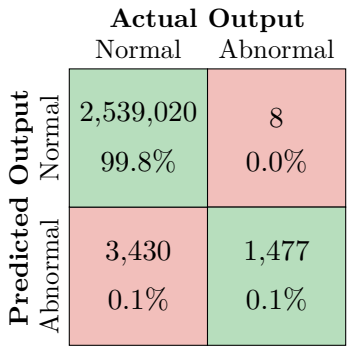


(a) Training Confusion Matrix

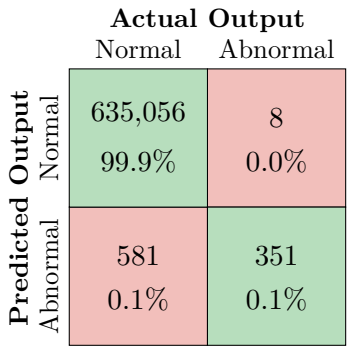


(b) Testing Confusion Matrix

Fig. B.40 The intelligent filtering technique evaluation confusion matrices for the Asus Netbook

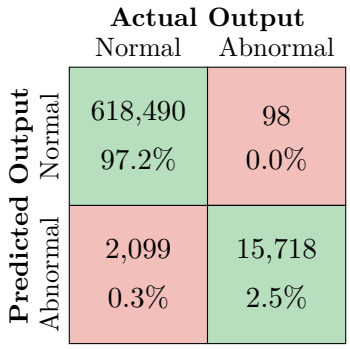


(a) Training Confusion Matrix

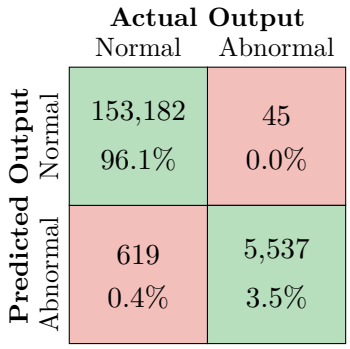


(b) Testing Confusion Matrix

Fig. B.41 The intelligent filtering technique evaluation confusion matrices for the Gateway Netbook

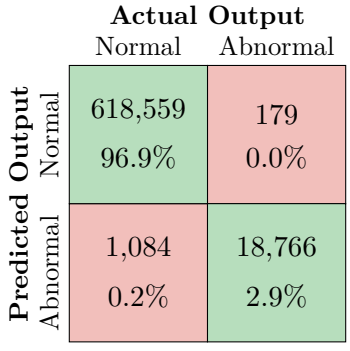


(a) Training Confusion Matrix

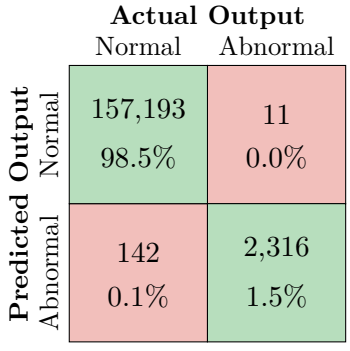


(b) Testing Confusion Matrix

Fig. B.42 The intelligent filtering technique Evaluation Confusion Matrix for Google Phone

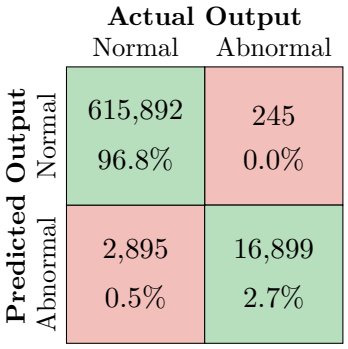


(a) Training Confusion Matrix

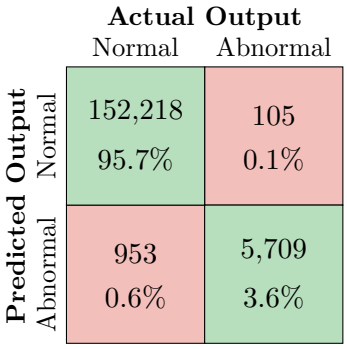


(b) Testing Confusion Matrix

Fig. B.43 The intelligent filtering technique evaluation confusion matrices for the Lenovo Laptop



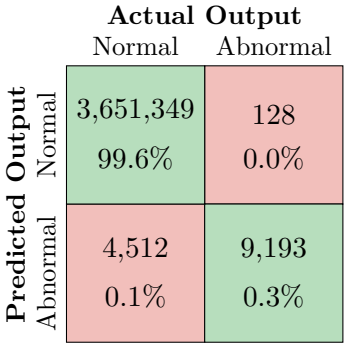
(a) Training Confusion Matrix



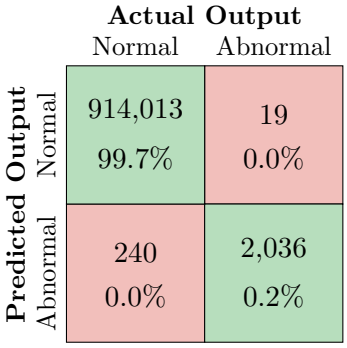
(b) Testing Confusion Matrix

Fig. B.44 The intelligent filtering technique Evaluation Confusion Matrix for Asus Tablet

B.4.2 Evaluation results for Isolated Network traffic Datasets

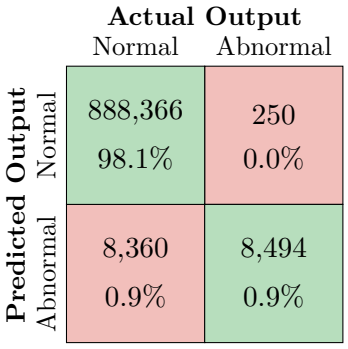


(a) Training Confusion Matrix

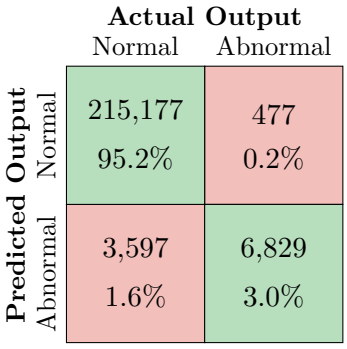


(b) Testing Confusion Matrix

Fig. B.45 The intelligent filtering technique Evaluation Confusion Matrix for iPad

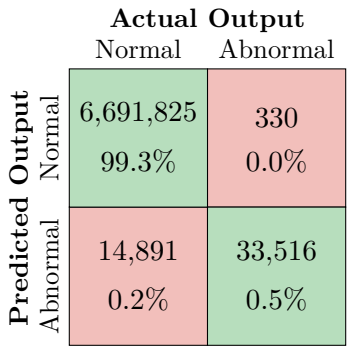


(a) Training Confusion Matrix

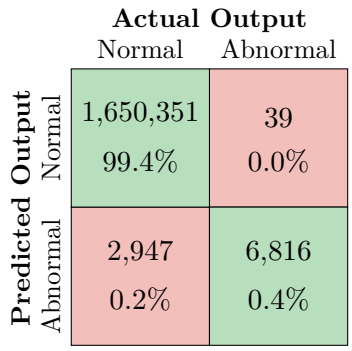


(b) Testing Confusion Matrix

Fig. B.46 The intelligent filtering technique Evaluation Confusion Matrix for iPhone 3G

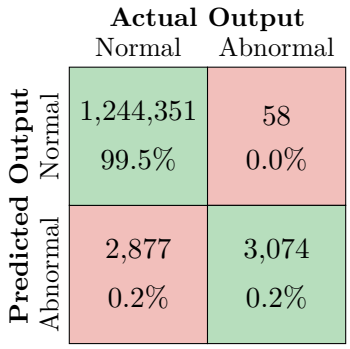


(a) Training Confusion Matrix

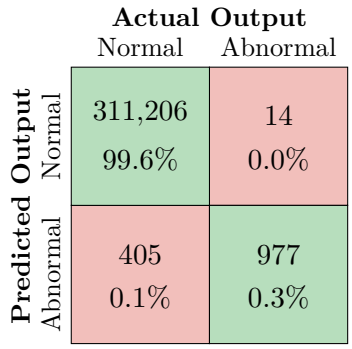


(b) Testing Confusion Matrix

Fig. B.47 The intelligent filtering technique evaluation confusion matrices for the iPhone 4G



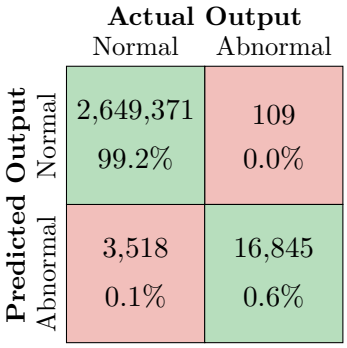
(a) Training Confusion Matrix



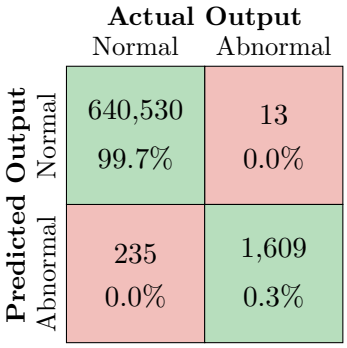
(b) Testing Confusion Matrix

Fig. B.48 The intelligent filtering technique Evaluation Confusion Matrix for Nokia Phone

B.4.3 Evaluation results for Passive Network Traffic Datasets

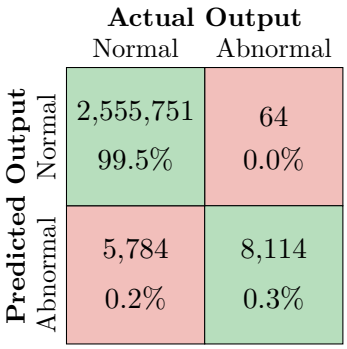


(a) Training Confusion Matrix

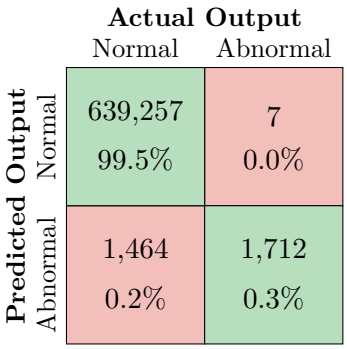


(b) Testing Confusion Matrix

Fig. B.49 The intelligent filtering technique Evaluation Confusion Matrix for Acer Netbook

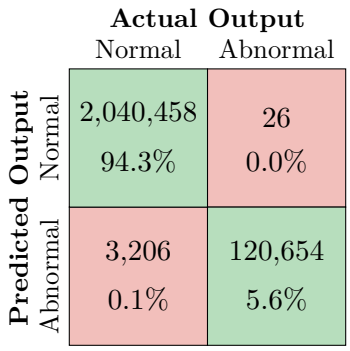


(a) Training Confusion Matrix

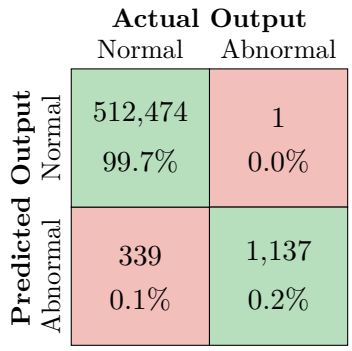


(b) Testing Confusion Matrix

Fig. B.50 The intelligent filtering technique Evaluation Confusion Matrix for Asus Netbook

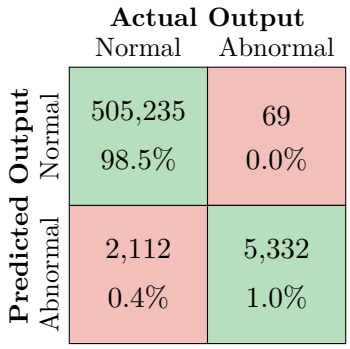


(a) Training Confusion Matrix

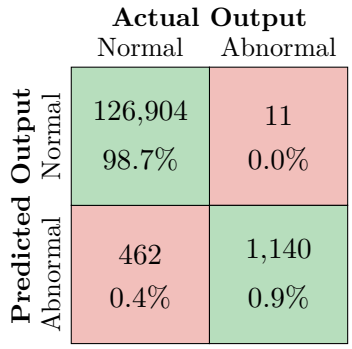


(b) Testing Confusion Matrix

Fig. B.51 The intelligent filtering technique evaluation confusion matrices for the Gateway Netbook

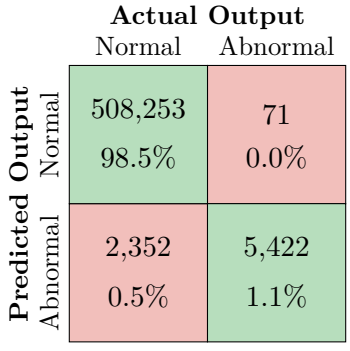


(a) Training Confusion Matrix

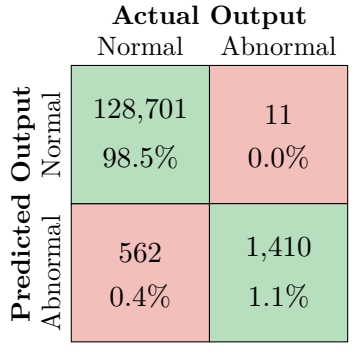


(b) Testing Confusion Matrix

Fig. B.52 The intelligent filtering technique evaluation confusion matrices for the Google Phone

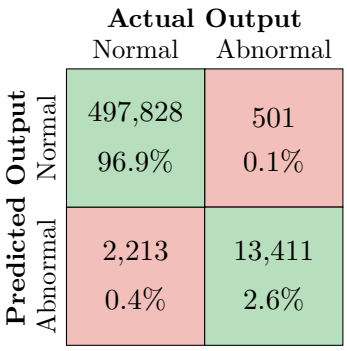


(a) Training Confusion Matrix

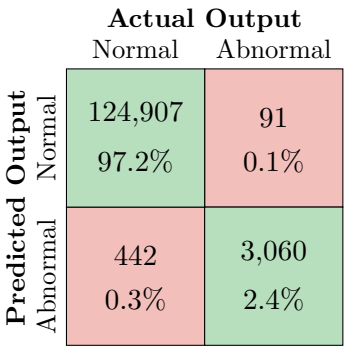


(b) Testing Confusion Matrix

Fig. B.53 The intelligent filtering technique Evaluation Confusion Matrix for Lenovo Laptop



(a) Training Confusion Matrix



(b) Testing Confusion Matrix

Fig. B.54 The intelligent filtering technique Evaluation Confusion Matrices for the Asus Tablet