UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

SHORT-TERM TIME SERIES ANALYSIS AND PREDICTION FOR ANTICIPATORY NETWORKING

TESIS PARA OPTAR AL GRADO DE
DOCTOR EN COMPUTACIÓN

DIEGO IGNACIO MADARIAGA ROMÁN

PROFESOR GUÍA:
BENJAMÍN BUSTOS CÁRDENAS

PROFESOR CO-GUÍA:
JAVIER BUSTOS JIMÉNEZ

MIEMBROS DE LA COMISIÓN:
ANDRÉS ABELIUK KIMELMAN
JOSÉ MIGUEL PIQUER GARDNER
PEDRO CASAS HERNÁNDEZ

Este trabajo ha sido parcialmente financiado por NIC Chile Research Labs y
ANID Doctorado Nacional 2019 - 21190450

SANTIAGO DE CHILE
2023

Resumen

ANÁLISIS Y PREDICCIÓN DE SERIES TEMPORALES A CORTO PLAZO PARA REDES ANTICIPADAS

Analizar y predecir el estado de la red ha sido de gran interés para la comunidad de redes a lo largo de toda su evolución, especialmente en la actualidad, debido a la orientación hacia redes autónomas mediante políticas adaptativas y de autoaprendizaje. Dada la alta periodicidad de los datos relacionados con redes de comunicaciones, múltiples estudios han abordado los problemas de detección de anomalías, detección de cambios de concepto y predicción para el manejo de redes. Sin embargo, estos trabajos se limitan en su mayoría a la teoría y no tienen en cuenta algunos retos primordiales para llevar a cabo implementaciones en el mundo real. En el siguiente trabajo de tesis, estudiamos distintos métodos de anticipación en redes, con el fin de mejorar la predicción de series temporales, manejando correctamente las anomalías presentes en la red. Concretamente, se abordan dos problemas esenciales en la anticipación del estado de la red, relacionados con (1) la detección de anomalías en el tráfico del Sistema de Nombres de Dominio (DNS) y (2) la predicción de la calidad de servicio en redes móviles.

El Sistema de Nombres de Dominio (DNS) es un componente crítico de la infraestructura de Internet, ya que prácticamente todas las actividades en Internet comienzan con una consulta DNS. Dada su importancia, cada vez hay mayor inquietud respecto a su vulnerabilidad ante ataques y fallos, puesto que estos pueden afectar negativamente a todos los recursos basados en Internet. Por ello, la detección de estos eventos es crucial para preservar el correcto funcionamiento de todos los componentes de este sistema, como los grandes Servidores de Nombres para dominios de primer nivel (TLD). Este trabajo presenta un método de detección de anomalías basado en predicciones (AD-BoP) el cual opera próximo al tiempo real y proporciona una metodología útil y fácilmente explicable para detectar anomalías en el tráfico DNS. Nuestro método se basa en la predicción de las estadísticas del tráfico DNS, y podría ser especialmente útil para que los operadores preserven la fiabilidad de sus servicios DNS. Tras un análisis exhaustivo, se demuestra que AD-BoP mejora el estado actual respecto a la detección de anomalías en servidores de nombres TLD autoritativos.

En cuanto a la calidad del servicio móvil, el rendimiento de red (*throughput*) se ha convertido en uno de los principales indicadores de desempeño. De hecho, a medida que las redes móviles evolucionan hacia nuevas tecnologías, el rendimiento de los usuarios se convierte cada vez en un indicador de desempeño más crucial, ya que múltiples aplicaciones de red dependen de su correcta predicción. A diferencia de la teoría, la experimentación empírica revela que, en la práctica, no existe una correlación directa entre el rendimiento del usuario y la calidad del canal. Por tanto, nos centramos en comprender mejor esta relación empírica para mejorar la predicción del rendimiento en redes móviles. En este trabajo, realizamos un estudio exhaustivo sobre el efecto práctico de la Relación señal a interferencia más ruido (SINR) en el rendimiento del usuario. Con este fin, proponemos y validamos un modelo probabilístico dependiente de SINR para estimar la distribución de probabilidad del rendimiento de los usuarios. Luego, presentamos dos métodos para predecir el rendimiento de forma fácilmente explicable, basados en nuestro modelo probabilístico. Estos métodos son de gran valor, ya que logran un desempeño excepcional en una amplia variedad de escenarios, basándose en una sola métrica contextual, la cual es obtenida directamente del dispositivo del usuario.

Abstract

Analyzing and predicting the network state has been of great interest to the networking community throughout all network evolution, especially nowadays, due to the calling to autonomous networks through adaptive and self-learning policies and self-evolution. Given the high periodicity present in network-related data, several works have addressed the problems of anomaly detection, concept drift detection, and prediction for networking. Nevertheless, these works are mostly confined to theory and do not consider some critical challenges for real-world implementations. In the following thesis work, we seek to study anticipatory networking methods to improve time series prediction, handling network anomalies correctly. Concretely, this thesis addresses two essential anticipatory networking problems related to (1) anomaly detection in Domain Name System (DNS) traffic and (2) prediction of mobile network Quality of Service (QoS).

The Domain Name System (DNS) is a critical component of Internet infrastructure, as almost every activity on the Internet starts with a DNS query. Given its importance, there is increasing concern over its vulnerability to attacks and failures, as they can negatively affect all Internet-based resources. Thus, detecting these events is crucial to preserve the correct functioning of all DNS components, such as high-volume name servers for top-level domains (TLD). This work presents a near real-time Anomaly Detection Based on Prediction (AD-BoP) method, providing a useful and easily explainable methodology to effectively detect DNS anomalies. AD-BoP is based on the prediction of expected DNS traffic statistics, and could be especially helpful for TLD registry operators to preserve their services' reliability. After an exhaustive analysis, AD-BoP is shown to improve the current state-of-the-art for anomaly detection in authoritative TLD name servers.

Regarding mobile QoS, user throughput has gained attention as one of the most relevant key performance indicators. Indeed, as mobile networks evolve towards newer technologies, user throughput becomes a more crucial performance indicator as a number of networking applications rely on its prediction. Different from theory, empirical experimentation reveals that, in practice, there is no direct correlation between user throughput and channel quality. Therefore, we focus on further understanding this empirical relationship in order to improve throughput prediction in mobile networks. In this work, we conduct a comprehensive study on the practical effect of the signal-to-interference-plus-noise ratio (SINR) on user throughput. We proposed and validated a novel SINR-dependent probabilistic mixture model for estimating the probability distribution of user throughput. Then, we present two easily explainable throughput prediction approaches based on our developed probabilistic mixture model. These approaches are valuable as they achieve outstanding performance in a wide range of scenarios, relying on a unique contextual metric obtained from the user equipment.

To my beautiful children

Acknowledgments

Several people played a decisive role during my Ph.D. journey, and therefore, I would like to express my deepest appreciation to all of them within the following lines.

As I couldn't have done this work without them, I'd like to recognize my advisors, Benjamín and Javier, for their helpful guidance and insightful suggestions. Particularly, I owe my thanks to Javier for providing me with work (at such an early stage), and networking opportunities I never thought I would find.

Many thanks to my NIC Labs family, which has been an essential part of my life for the last eight years. I'll keep in my heart all those beautiful and happy moments we lived together. Thank you for embracing me with open arms, supporting my goals, and being my home for so long.

Particularly helpful to me during this time were Martín, Lucas, and Magaly, who directly contributed to the results of this manuscript. Thanks for all those working hours and joint effort, which in the end had good rewards.

Also, I must thank my parents, Edgar and Judith, and siblings, Andrea, Samuel, Pablo, and Javier. I'm deeply indebted to all of you for never wavering in your support and never letting me down.

Of course, the culmination of this work would not have been possible without the support and nurturing of my life partner, Catalina. Thanks for your love and company throughout all these years. Sometimes life can be deceiving, but it's always better when we're together.

Last but not least, special thanks to my three beautiful children, whose help cannot be overlooked. Iván, Pedro, and Eloísa, you are the reason for my smile and happiness. There's no biggest number, no farthest star to my love for you.

I'd like to close this chapter in my life using the same words my grandfather Álvaro (at peace now) gave me ten years ago, right before starting my undergraduate studies: *Every good gift and every perfect gift is from above, and cometh down from the Father of lights, with whom is no variableness, neither shadow of turning.*

Table of Content

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem statement

Anticipatory networking is a recent branch of network optimization based on the prediction of the system state, which is assumed to be, to some degree, predictable [25]. Since most network data can be naturally represented as a time series, several works base their network predictions on time series forecasting techniques, where the high periodicity of network data is often exploited to increase accuracy. Nevertheless, most anticipatory networking studies are confined to theory and do not consider some challenges for real-world implementations, such as the need for online prediction methods, the lack of large amounts of data to train the forecast models, and the need to update their predictions against the presence of anomalies or concept drifts in real time series. Accordingly, there is great interest in **developing *online* prediction methods for anticipatory networking, being able to handle network anomalies correctly**. In this thesis work, two anticipatory networking problems will be addressed, in order to achieve real-world solutions. These two networking problems focus on different network layers in the seven-layer OSI model of computer networking, evidencing the comprehensive scope of the methodology employed in this work across the network.

Firstly, we consider an anticipatory networking problem related to the topmost layer in the OSI model, i.e., the closest to the end-user (layer 7). In particular, we focus on one of the most crucial application layer protocols: the Domain Name System (DNS) protocol. The Domain Name System (DNS) is a critical component of Internet infrastructure, as almost every activity on the Internet starts with a DNS query. Given its importance, there is increasing concern over its vulnerability to attacks and failures, as they can negatively affect all Internet-based resources. Thus, detecting these events is crucial to preserve the correct functioning of all DNS components, including high-volume name servers for top-level domains (TLD). In this part of the thesis work, we focus on the following problem: *to develop an easily explainable methodology to detect DNS anomalies in TLD name servers effectively.*

Secondly, we focus on the anticipatory networking problem of predicting mobile network QoS, which is of particular interest for the delivery of multimedia content. For this purpose, we examine the physical layer (layer 1), which deals with bit-level transmission. From a

theoretical perspective, some channel quality metrics, such as the received signal strength and the signal-to-noise ratio, can help determine the channel capacity. Therefore, we claim that a correct understanding and use of physical layer information could be helpful to estimate some QoS indicators (e.g. throughput). In this part of the thesis work, we focus on the following problem: *to develop an explainable methodology to predict throughput in cellular networks by following a probability distribution approach.*

1.2 Research objectives

1.2.1 General objective

To develop anticipatory networking models to improve the performance of real prediction-based networking problems, being able to handle network anomalies correctly.

1.2.2 Specific objectives

For each of the two anticipatory networking problems mentioned in Section 1.1, the following specific objectives are defined:

A. To obtain a proper representation of the data to be analyzed and predicted.

B. To establish accuracy measures to be used for comparison between different prediction models.

C. To clearly identify the limits of predictability for each problem.

D. To develop a predictive model that considers network anomalies to adapt its predictions.

1.3 Hypothesis

It is possible to improve the accuracy of short-term prediction in real anticipatory networking problems, by using online time series analysis and its adaption against anomalies and concept drifts in the data.

1.4 Methodology

For both anticipatory networking problems related to (1) anomaly detection in DNS traffic and (2) prediction of mobile network QoS, the following methodology will be applied:

1. Conduct a literature review to analyze the background related to the research question, clearly identifying the research gaps to be addressed by the thesis work.
2. Select the appropriate data sources to study each anticipatory networking problem.
3. Obtain a proper data representation of the collected data, using a proper aggregation method for raw data, and choosing an appropriate dimensionality.
4. Select the accuracy measures to be used for comparison between different prediction models.
5. Implement state-of-the-art baseline algorithms for each anticipatory networking problem.
6. Develop novel prediction models for each anticipatory networking problem, considering the presence of network anomalies.
7. Evaluate the prediction accuracy of state-of-the-art algorithms against the proposed models.
8. Verify the hypothesis stated in Section 1.3.

1.5 Contributions of the thesis work

Regarding the first anticipatory networking problem being studied (related to DNS traffic), we present an important contribution to the field of DNS anomaly detection, providing a useful and easily explainable methodology to detect DNS anomalies (AD-BoP method), based on the expected DNS traffic statistics. The proposed method could be especially helpful for TLD registry operators to preserve the reliability of their services.

Additionally, we designed a useful tool to simulate anomalous traffic and inject it into real DNS data. These artificially created DNS anomalies are customizable in terms of the duration and magnitude of the attacks to provide different scenarios to test. The designed tool is able to simulate some well-known attacks to the DNS infrastructure. In particular, this open-source tool was employed to evaluate the performance of the proposed anomaly detection methodology against different baseline models.

Regarding the second anticipatory networking problem being studied (related to mobile QoS), we present a novel measurement methodology for Android devices capable of passively monitoring Internet traffic in user-space, and providing a comprehensive set of contextual information. Our passive approach relies on the implementation of a local VPN server residing inside the client device to manage all Internet traffic and obtain crucial information about network flows. Our proposed methodology is advantageous to collect inexpensive crowdsourcing measurements from a medium-sized set of mobile users, and yet to obtain valuable data regarding the target population.

In addition, we thoroughly analyze the impact of channel quality variations on user throughput using a novel approach. Different from other authors, we study the user throughput as a random variable that depends on the current signal-to-interference-plus-noise ratio

(SINR). Thus, we model the distribution of user throughput as an SINR-dependent probabilistic mixture model that properly fits the empirical data. This approach allows a more comprehensive understanding of the empirical effect of the SINR on user throughput, as we could directly apply different concepts of probability distribution theory. Moreover, we present a methodology to extrapolate the probability distribution of user throughput for any SINR value, even for those absent (or poorly represented) in the original dataset.

Then, we present two different approaches to predict instantaneous user throughput based on our SINR-dependent probabilistic mixture model. These methodologies use two strategies to estimate the mathematical expectation of the random variable (user throughput). Therefore, our throughput estimators can be directly explainable as they correspond to the expected value of the probability distribution of user throughput. According to our experimental results, these approaches are shown to be valuable for practical throughput prediction applications at different time scales.

Finally, we perform a rigorous examination of the misuse of log-scaled signal strength values, which is a common issue within the literature on mobile computing. We present the physical and mathematical formalities of how signal strength values must be handled in a scientific environment. Furthermore, we present a solution to the difficulties of aggregating signal strength in mobile crowdsourcing scenarios, such as the low number of measurements and nonuniformity in spatial distribution.

1.5.1 Resulting publications

This thesis work has resulted in the following peer-reviewed publications:

- **Diego Madariaga**, Javier Madariaga, Martin Panza, Javier Bustos-Jimenez, and Benjamin Bustos. Detecting Anomalies at a TLD Name Server Based on DNS Traffic Predictions. *IEEE Transactions on Network and Service Management 18(1):1016-1030.* IEEE, 2021. [92]

  Summary: We presented a methodology for Anomaly Detection Based on Prediction (AD-BoP), by using a machine learning model to forecast different portions of the whole DNS traffic. Our approach was demonstrated to improve the state-of-the-art in DNS anomaly detection for top-level domain (TLD) name servers, providing a useful and easily explainable methodology, which could be especially helpful for TLD registry operators to preserve the reliability of their services.

- **Diego Madariaga**, Javier Madariaga, Javier Bustos-Jimenez, and Benjamin Bustos. Improving signal Strength Aggregation for Mobile Crowdsourcing Scenarios. *Sensors 21(4):1084.* MDPI, 2021. [91]

  Summary: We proposed an improvement of the signal strength aggregation with a special focus on mobile crowdsourcing, by studying the physical and mathematical formalities of how signal strength values must be handled in a scientific environment, and by presenting a novel aggregation method. Our proposed method was shown to

properly deal with the difficulties of aggregating signal strength in mobile crowdsourcing scenarios, such as a low number of measurements and nonuniformity in spatial distribution.

- **Diego Madariaga**, Lucas Torrealba, Javier Madariaga, Javier Bustos-Jiménez, and Benjamin Bustos. PePa Ping Dataset: Comprehensive Contextualization of Periodic Passive Ping in Wireless Networks. In *Proc. 12th ACM Multimedia Systems Conference (MMSYS'21),* pages 274-280. ACM, 2021. [96]

  Summary: In this work, we presented a useful dataset collected by a novel passive measurement methodology for mobile devices. Our methodology is able to collect valuable information regarding network usage, empirical quality of service, and a comprehensive set of mobile user's contextual information. Thus, we provided a useful source that can be of great importance to a variety of studies related to network usage behavior and network performance.

- **Diego Madariaga**, Lucas Torrealba, Javier Madariaga, Javiera Bermúdez, and Javier Bustos-Jiménez. Analyzing the Adoption of QUIC From a Mobile Development Perspective. In *Workshop on Evolution, Performance, and Interoperability of QUIC (EPIQ '20).* ACM, 2020. [95]

  Summary: In this work, we used a mobile network dataset collected by a novel passive measurement methodology to study the real usage of different network protocols. Particularly, we analyze the adoption of QUIC (Quick UDP Internet Connections) as the transport protocol utilized by mobile applications for accessing the web.

Additionally, this thesis work has been partially related to the following peer-reviewed publications:

- **Diego Madariaga**, Martín Panza, and Javier Bustos-Jiménez. I'm only unhappy when it rains: Forecasting mobile QoS with weather conditions. In *Network Traffic Measurement and Analysis Conference (TMA '18).* IEEE, 2018. [93]

  Summary: In this work, we studied the feasibility of forecasting mobile signal strength using crowdsourced measurements. We proposed different methods to forecast signal strength based on time series analysis and considering external information about weather conditions such as temperature, humidity, and precipitations. According to our results, including weather information improves the accuracy of signal strength forecast models.

- **Diego Madariaga**, Martín Panza, and Javier Bustos-Jiménez. DNS traffic forecasting using deep neural networks. In *Machine Learning for Networking: First International Conference, (MLN '18).* Springer, 2019. [94]

  Summary: We studied the feasibility of predicting DNS traffic using machine learning models for time series forecasting. The proposed forecasting models were shown to properly capture the traffic patterns generated by users' DNS queries. The prediction of DNS traffic has a huge relevance since a big difference between the expected DNS traffic and the real one, could be a sign of an anomaly in the data stream caused by an attack or a failure.

- Martín Panza, **Diego Madariaga**, and Javier Bustos-Jiménez. Extracting human behavior patterns from DNS traffic. In *Annals of Telecommunications 77(5-6), 407-420.* Springer, 2022. [126].

  Summary: In this work, we analyzed the effect of human patterns and users' activities in DNS traffic. We thoroughly inspected real DNS data to better understand the detection of trends and patterns regarding Internet usage, which can be particularly helpful for analyzing human behavior.

- Panagiota Katsikouli, **Diego Madariaga**, Aline Carneiro Viana, Alberto Tarable, and Marco Fiore. DuctiLoc: Energy-Efficient Location Sampling With Configurable Accuracy. *IEEE Access 11 (2023): 15375-15389.* IEEE, 2023. [72]

  Summary: We presented a location sampling mechanism that dynamically adapts its position sampling frequency to individual mobility habits and target accuracy level. The proposed approach, is energy efficient, as it does not rely on power-hungry sensors or expensive computations; moreover, it provides a handy knob to control energy usage, by configuring the target positioning accuracy. Controlling the trade-off between accuracy and sampling rate of human movement is useful in a number of contexts, including mobile computing and cellular networks.

Chapter 2

Anticipatory Networking for DNS

2.1 Introduction

With the continuous growth of the Internet and its increasing number of users during the last decades, the Domain Name System (DNS) has also experienced an important evolution. DNS has become a vital part of the Internet itself, as it is responsible for the translation between domain names and IP addresses. This means that almost every activity on the Internet starts with a DNS query (and often several) [19]. Along with the importance and evolvement of DNS, there is increasing concern over its vulnerability to attacks and failures. Consequently, multiple mitigation strategies have been raised to reduce the risk of a negative impact on all Internet-based resources [17].

As one of the main requirements on the DNS is its availability [180], a considerable amount of literature has been published on the topic of automatic DNS anomaly detection. These anomalies are commonly defined as traffic patterns that do not conform to expected normal behavior [30, 45, 2]. Identifying these unexpected DNS patterns is important, as detecting irregularities in the system's behavior could inform about abnormal or malicious events.

Additionally, as DNS is a distributed hierarchical database, top-level DNS servers are more likely to be the entry of DNS query behavior of clients [172]. Therefore, TLD name servers are more appropriate to be indicators of network anomalies since these anomalies can be reflected on their DNS traffic.

Hence, a system that fulfills the goal of automatic anomaly detection could be of great benefit for TLD registry operators, since offering its users a reliable DNS service is crucial for their business. Early detection of threats would allow them to take quick action, in order to preserve the system's correct functioning.

The problem of DNS anomaly detection is, however, a challenging task, as DNS traffic naturally evidences some well-known challenges for general anomaly detection methodologies [30]:

1) Defining a normal DNS traffic behavior that encompasses every possible normal behav-

ior is very difficult. In addition, the boundary between normal and anomalous traffic behavior is not precise.

2) When DNS anomalies are the result of malicious actions (attacks), the adversaries could try to adapt themselves to make the anomalous DNS traffic appear normal.

3) DNS traffic behavior can frequently evolve due to the evolution of Internet data usage or changes in the system's configurations. Thus, a current notion of normal DNS traffic behavior might not be sufficiently representative in the future.

4) DNS traffic statistics measured in distinct scenarios can be significantly different from each other. Thus, applying a technique developed in one scenario to another is not straightforward.

5) There is a lack of labeled data for the training/validation of DNS anomaly detection techniques.

6) DNS traffic data can contain random noise that tends to be similar to the actual traffic anomalies and hence is difficult to distinguish.

This work presents a methodology for Anomaly Detection Based on Prediction (AD-BoP), by using a machine learning model to forecast different portions of the whole DNS traffic. The proposed method offers an effective mechanism to detect anomalies in the traffic of top-level domain name servers, as it was designed to face the six previously mentioned challenges successfully:

1) By using a learning approach, the model can be fed with a large amount of historical DNS data. Thus, the model defines a normal behavior from a large number of normal DNS traffic examples.

2) Even when DNS attacks can try to stay hidden when analyzing the overall DNS traffic, they will inevitably impact some portion of the traffic. Thus, by analyzing different portions of the traffic separately, AD-BoP enhances the detection of these malicious anomalies.

3) By using a learning approach, the AD-BoP method is continuously updated and adapted to the evolution of DNS traffic behavior. Thus, the notion of normal DNS traffic is also being updated.

4) The presented anomaly detection strategy only depends on the behavior of the scenario where it is deployed. Therefore, AD-BoP can be applied to different DNS scenarios as it will learn the normal behavior in each case. This study shows the satisfactory performance of the AD-BoP method when applied to significantly different scenarios of real TLD name servers.

5) The lack of labeled data is not a problem for AD-BoP, since it uses an unsupervised learning approach. Consequently, AD-BoP is also able to detect new unknown anomalous events.

6) Other works that strictly focus on network attacks consider the detection of noisy traffic as a false positive [48]. By contrast, this study does not consider that the detection of noisy traffic is incorrect per se, since it can be actually produced by other causes, such as misconfigurations or system failures, causing a malfunction of the TLD nameservers.

For a better evaluation of the proposed AD-BoP method, this study compares its performance against two state-of-the-art methodologies for DNS anomaly detection in TLD name servers. As a result, AD-BoP is shown to present some improvements to the state-of-the-art methods:

- AD-BoP presents a simpler interpretation of the sensitivity parameter (threshold) selected.
- AD-BoP presents a direct interpretation of detected anomalies, which could help TLD operators rapidly find out the real cause of the abnormal behavior.
- AD-BoP presents a better performance in detecting some well-known DNS attacks, clearly separating those attacks from normal DNS behavior.

The results in this chapter present an important contribution to the field of DNS anomaly detection, providing a useful and easily explainable methodology to detect DNS anomalies, based on the expected DNS traffic statistics. The proposed method could be especially helpful for TLD registry operators to preserve the reliability of their services.

2.2 Background

A considerable amount of literature has been published on detecting DNS anomalies using a wide variety of methodologies. Some research studies proposed methods to detect some specific DNS attacks, such as Domain Fluxing [177], Botnet Domains [166], and Kaminsky Cache Poisoning [112]. In this work, we propose an unsupervised learning approach to perform DNS anomaly detection, and therefore, our methodology is able to detect a wider variety of both known and unknown DNS anomalous events.

DNS anomaly detection can be roughly divided into two categories regarding where the DNS traffic is gathered. On the one hand, some works use probes installed in a particular network to gather DNS traffic, e.g., from university campus networks [29, 180]. On the other hand, some research studies analyze the DNS traffic directly on DNS servers without additional monitoring infrastructure. The presented study uses the latter approach, by detecting DNS traffic anomalies in TLD name servers.

As TLD name servers are more likely to be indicators of network anomalies [172], a number of authors have reported traffic analyses for different TLD name servers. Wang et al. [172] used data from the Chinese .cn ccTLD to analyze the impact of a major national event on the DNS traffic behavior. In addition, Deri et al. [35] proposed a methodology for the Italian .it ccTLD for permanent DNS traffic monitoring. Researchers from InternetNZ,

the registry for the .nz ccTLD for New Zealand, presented a time series approach to analyze trends on DNS queries and to inspect the presence of anomalies in historical DNS traffic [133]. Closer to the goals of this work, Mikkle et al. [102] and Robberechts et al. [143] addressed the automatic anomaly detection for a TLD name server, using data from the .cz ccTLD for the Czech Republic and the .be ccTLD for Belgium, respectively. These two state-of-the-art anomaly detection methodologies, denominated as CZ.NIC method [102] and QLAD [143], serve as a baseline for our proposed approach.

Authors from CZ.NIC, the administrator of the .cz ccTLD for Czech Republic, implemented a profile-based anomaly detection methodology for extracting hidden anomalies in DNS traffic [102]. This method performs as follows:

1. All DNS packets inside a time window are represented by a tuple (t, A), where t is the arrival timestamp of the packet and A is a packet identifier. This identifier can be the source IP address or the first domain name in the DNS query. These two possible policies for A are referred to as source IP address policy and query name policy.

2. All A identifiers are hashed using N independent $k-$universal hash functions (h_n), with hash table size equal to M. Thus, each hash function h_n splits the original trace into M sub-traces.

3. The method computes several sets denoted by $X_{n,m}$, containing all packets in which the n^{th} hash function mapped their A identifier to m.

4. Then, the sets $X_{n,m}$ are aggregated jointly over a collection of aggregation levels j with size J to form the $X^j_{n,m}(t)$ time series.

5. Each $X^j_{n,m}(t)$ is modeled using Gamma distribution laws with parameters α (shape) and β (scale). For every $X^j_{n,m}(t)$, α and β parameters are referred to as $(\alpha^j_{n,m}, \beta^j_{n,m})$.

6. For every hash function h_n, this method estimates the standard sample means with respect to m (α^j_n and β^j_n), and their variances ($\sigma^2_{n,\alpha,j}$ and $\sigma^2_{n,\beta,j}$).

7. Considering the Mahalanobis distance for $\gamma \in \{\alpha, \beta\}$ defined as

$$(D_{n,m})^2 = \frac{1}{J} \sum_{j=1}^{J} \frac{(\gamma^j_{n,m} - \gamma^j_n)^2}{\sigma^2_{n,\gamma,j}} \tag{2.1}$$

   and considering the threshold value λ, if $D_{n,m} > \lambda$, then the set $X_{n,m}$ is said to contain at least one anomaly.

8. Finally, the intersection of all $X_{n,m}$ such that $D_{m,n} > \lambda$, corresponds to the detected anomalies within the scanned time window.

Several limitations to this method need to be acknowledged:

- One major drawback is that there is no clear interpretation of its threshold parameter, as it is applied after a complex data processing of the DNS traffic being analyzed.

- This method is supposed to fail to detect attacks that use randomly spoofed IP addresses because these malicious packets will belong to different sets [143]. This is an important limitation since the use of spoofed IP addresses is a widely used strategy for performing DNS attacks.

In addition, Robberechts et al. [143] used data belonging to the .be ccTLD for Belgium to present the implementation of their DNS anomaly detection system: QLAD. As their authors commented, it is a proof-of-concept system, which is divided in two subsystems named QLAD-global and QLAD-flow.

QLAD-global is a method based on the fact that DNS traffic anomalies will change the normal distribution of one or more traffic features. Thus, the detection of anomalies can be performed by detecting anomalies in the entropy of these traffic features. This method takes into account the following DNS traffic features:

- The Number of DNS queries for each second-level domain name (1)
- The Number of DNS queries for each record type (2)
- The Number of DNS responses for each response code (3)
- The number of requests by each client (4), ASN (5), and country (6).
- DNS response sizes (7).

QLAD-global considers the division of the DNS traffic in successive windows of time T. At each time interval, the entropies of the traffic features are calculated by using the following equation:

$$H(X) = -\frac{1}{\log(n)} \sum_{i=1}^{n} p(x_i) \log(p(x_i)) \tag{2.2}$$

where X is a traffic feature that can take values $x_1, ..., x_n$ with probability mass functions $p(x_1), ..., p(x_n)$. The division by $\log(n)$ (the maximum entropy) is adopted to avoid the effect of DNS traffic periodicity on the measured entropy, as proposed by Nychis et al. [151].

At each time window, the system computes the actual entropy of each traffic feature, denoted by e_t. Then, QLAD-global uses e_t to update an exponential moving average (EMA_t) and an exponential moving standard deviation (EMS_t). Finally, an anomaly is reported if the following is satisfied by any traffic feature:

$$\frac{|e_t - \text{EMA}_t|}{\text{EMS}_t} > \lambda \tag{2.3}$$

where λ is the threshold that controls the sensitivity of the anomaly detection.

However, this method has a number of limitations for a real-world implementation:

- External databases must be maintained to perform geolocation of IP addresses. This is required to compute the number of requests by each ASN and country, whose information cannot be obtained directly from DNS packets.
- Threshold λ cannot be directly set, given the lack of prior knowledge about the expected entropy values. In addition, λ values do not have a direct interpretation regarding DNS traffic, hindering its understanding.
- The detection of an anomaly in the entropy of a traffic feature does not necessarily give enough information to a TLD operator to find out the real cause of the anomaly.

QLAD-flow method takes the previously mentioned work from CZ.NIC as a basis, with the addition of a new third policy named ASN policy, which is a generalization of the source IP address policy. However, in their experimental results, the ASN policy did not provide newer insights to the anomaly detection regarding the use of source IP policy. Therefore, only the QLAD-global subsystem of QLAD will be considered in this study, as QLAD-flow does not outperform the original CZ.NIC method.

2.3 AD-BoP Methodology

Our proposed Anomaly Detection Based on Prediction methodology (AD-BoP) is based on the fact that DNS anomalies will impact and change the expected traffic statistics for a given time interval. Thus, good predictions of these expected statistics can be used to detect anomalous traffic, according to the difference between real and expected values. Accordingly, this approach is well-aligned with the usual definition of an anomaly, defined as a pattern that does not conform to expected normal behavior [30, 45, 2].

In earlier research [94], me and coauthors presented the feasibility of forecasting DNS traffic volume by using data from an authoritative name server for a ccTLD. We found that recurrent neural networks (specifically Long Short Term Memory networks) performed better than other simpler statistical models in predicting DNS traffic. The well-working of LSTM networks was reflected in their ability to capture the periodic patterns in the traffic and detect some abrupt phase changes.

Even though this previous work only studied the prediction of DNS traffic, it serves as the basis for developing our proposed anomaly detection method. In the present work, we take advantage of our previous findings regarding the feasibility of predicting DNS traffic to develop an anomaly detection mechanism based on the prediction of normal DNS traffic. By predicting DNS traffic for a future interval, we can use the difference between this prediction and the real traffic to indicate the presence of an anomaly.

As the volume of DNS traffic at a TLD is intrinsically very large, some traffic anomalies could affect just a particular segment of the traffic and stay hidden when analyzing the overall DNS traffic. Therefore, we are not just interested in predicting the whole volume of DNS traffic to find differences between predicted and real values. We are also interested in

predicting some particular segments of DNS traffic, to analyze in more detail the presence of abnormal traffic behavior.

For that reason, we propose the aggregation of DNS data into different groups to make independent predictions. This approach could help TLD operators obtain more details about the specific sections of the whole DNS traffic affected by a given anomaly.

Given the aforementioned, our proposed AD-BoP method is focused on the following nine DNS traffic features:

- The number of DNS queries of types `A` (1), `AAAA` (2), `NS` (3), `MX` (4), and `ANY` (5).
- The number of unique queried domains (6).
- The number of DNS response packets with codes `NXDOMAIN` (7) and `NOERROR` (8).
- The total number of DNS packets (9).

AD-BoP considers a division of the DNS traffic in successive windows of time T. For each time interval, AD-BoP calculates a feature vector with the nine traffic features. Then, by consecutively joining these feature vectors, AD-BoP creates a multivariate time series, where each of its nine DNS-related features describes the behavior of a specific portion of the DNS traffic. Figure 2.1 shows the temporal behavior of four of the nine selected features for a real authoritative name server for the Chilean .cl ccTLD. The figure contains two days of real data, where the differences between each time series are clearly visible.

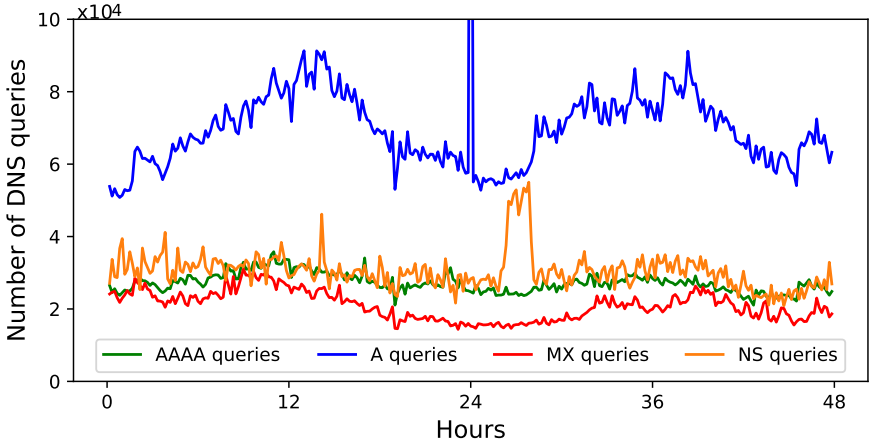


Figure 2.1: Differences in the behavior of 4 DNS traffic features at a TLD name server

Furthermore, Figure 2.1 illustrates a clear example of why the aggregation of DNS data into different groups can enhance the detection of anomalies, as some anomalies could impact only a specific segment of the DNS traffic. In the figure, around hour 24, there is a very pronounced anomaly in the number of DNS queries for `A` records (blue line), which is not visible at any of the other three time series. In addition, a second anomaly is visible around hour 26, with a duration of approximately three hours. This second anomaly only affects the number of DNS queries for `NS` records (orange line).

As suggested by our previous work [94], we used a forecast model based on LSTM networks to predict the time series regarding the nine selected DNS features. We modeled the problem as the prediction of one multivariate time series rather than the prediction of nine individual time series. Each DNS feature depends not only on its past values but also on other DNS variables because, together, they all represent the behavior of DNS clients. Thus, we found a multivariate time series analysis more appropriate.

2.3.1 LSTM Forecast Model

Artificial Neural Network models, loosely based on the natural neural networks of the human brain, consist of the interconnection of several nodes that individually define weights and operations to transmit data over their own connections and, thus, over the network. This transmission of data allows the network to learn from the input to give predictions of future information.

Recurrent Neural Networks (RNNs) are a class of neural networks that include loops within their connections as a mechanism to store information in their cells and make it persist through the steps of the training process. In this way, the output obtained by each neuron is influenced by both the new input and the values obtained from previous computations. This characteristic is useful when dealing with time series or sequential data, and consequently, RNNs are widely used for time series prediction [34, 32].

LSTM is a particular type of recurrent neural network designed to enhance the performance on Long-Term dependencies, as it deals better with the vanishing gradient problem related to the constant backpropagation process [56]. It achieves this goal by adding a new module to determine what information to store and forget.

The implementation of an LSTM unit consists of three gates: input gate, output gate, and forget gate that are related according to the following equations:

$$c_t = i_t \circ \tanh(W_c x_t + V_c y_{t-1} + b_c) + f_t \circ c_{t-1} \tag{2.4}$$

$$y_t = o_t \circ \tanh(c_t), \tag{2.5}$$

where i_t, f_t, o_t are the respective activation functions of each gate:

$$g_t = \sigma(W_g x_t + V_g y_{t-1} + b_g), \tag{2.6}$$

where σ is the sigmoid function, g is the corresponding gate, W and V are weight matrices, b is a bias vector, x_t and y_t are input and output vectors of the step t, and $\circ$ corresponds to the entry-wise product between two matrices.

2.3.2 Detection Strategy

After a given time interval $t-1$, AD-BoP employs the LSTM model to compute a prediction for the next interval t regarding the nine traffic features. Then, after the completion of interval t, AD-BoP obtains the nine real traffic features for interval t (from real DNS traffic), and uses them to update the LSTM model for the next prediction. AD-BoP will label the interval t as anomalous if the following is satisfied by any of the nine traffic features:

$$\frac{|y_t - y'_t|}{y'_t} > \lambda \tag{2.7}$$

where y'_t denotes the predicted value and y_t denotes the real value of a given traffic feature at interval t. The λ value is the threshold that controls the method's sensitivity and is defined as the maximum accepted relative change of the traffic feature, calculated from the absolute difference between expected and real values with reference to the expected value.

λ value may not necessarily be equal for all traffic features, and it can be independently set for each one. However, in the present work, we will always consider the same λ value for the nine traffic features to enhance the readability of our findings.

Unlike threshold values used for controlling the sensitivity in both anomaly detection methods explained in Section 2.2 (CZ.NIC method and QLAD-global), our threshold parameter can be directly interpreted in terms of DNS traffic, as shown in the following example for $\lambda = 0.6$:

"The number of DNS queries for nonexistent domains is more than 60% higher than expected."

This straightforward interpretation of detected anomalies could be very helpful for TLD operators to find out the real cause of the detected abnormal behavior.

2.4 DNS Anomaly Injection

One of the main limitations of testing the effectiveness of DNS server monitoring and troubleshooting tools is the lack of labeled anomalies in the historical traffic data from real DNS servers. This means that DNS anomaly detection methodologies cannot be evaluated using common metrics such as precision and recall.

To evaluate the proposed anomaly detection methodology, we designed a useful tool to simulate anomalous traffic and inject it into real DNS data (the source code is publicly available on `https://github.com/niclabs/dns-anomaly-injection`). This tool takes as input `.pcap` files with DNS traffic from a TLD authoritative name server, and returns new `.pcap` files with injected anomalies in addition to the original traffic. These artificially created DNS anomalies are customizable in terms of the duration and magnitude of the attacks to provide different scenarios to test. The designed tool is able to simulate some well-known attacks to the DNS infrastructure:

Random Subdomain: In this attack, many DNS queries are sent to the DNS server for a single target domain. These queries are created by adding randomly generated subdomains to the victim's domain. Random subdomain attacks cause the authoritative name servers of the target domain to experience DDoS, and responses may never come back from the target domain [47]. Even when the target of this attack is not the authoritative name server for the TLD, if the attack is distributed among many open resolvers, the evidence of the attack will also be visible on the TLD servers.

DNS Amplification: Like other amplification attacks, DNS Amplification is a type of reflection attack, where TLD nameservers are exploited as unknowing agents to perform the attacks [5]. In this case, the reflection is achieved by sending small DNS queries that result in large DNS responses to a spoofed IP address. DNS Amplification can be performed by using the EDNS0 DNS protocol extension, which allows for large DNS messages, or using the cryptographic feature of the DNS security extension (DNSSEC) to increase message size. In addition, this attack can be performed by using DNS record type `ANY`, which returns all records of all types known regarding the queried domain in a single request. The designed DNS anomaly injection tool implements this last type of DNS Amplification attack, which uses query type `ANY` to amplify DNS queries.

NXDOMAIN Flood: This is a type of DNS flood attack that attempts to overwhelm server resources and impact performance. It works by sending a flood of queries for nonexistent domain names to an authoritative name server. This attack causes the server's cache to fill up with `NXDOMAIN` results, slowing DNS server response time for legitimate requests [4].

2.5 Dataset Overview

To test the proposed AD-BoP method in a real-world environment, we used data collected directly by the official registry for the Chilean .cl ccTLD: NIC Chile (Network Information Center of Chile) [118]. NIC Chile maintains a network of name servers for the .cl ccTLD worldwide to provide a robust and stable service with excellent response times. The data used in this study consist of a month of normal operation traffic from two authoritative name servers under the control of NIC Chile, belonging to an anycast configuration along with other servers [117]. The first name server is located in Santiago, Chile, whereas the second name server is located in Amsterdam, Netherlands. In the following, the collected data from these two servers will be referred to as *Santiago* dataset and *Amsterdam* dataset.

The collected data from both servers start on 7 November 2018, until 6 December of the same year. For each server, the dataset is divided into 4180 `.pcap` files, each one containing 10-min of DNS traffic data. This represents a total of 480 GB of raw `.pcap` files (110 GB for the *Santiago* dataset and 370 GB for the *Amsterdam* dataset).

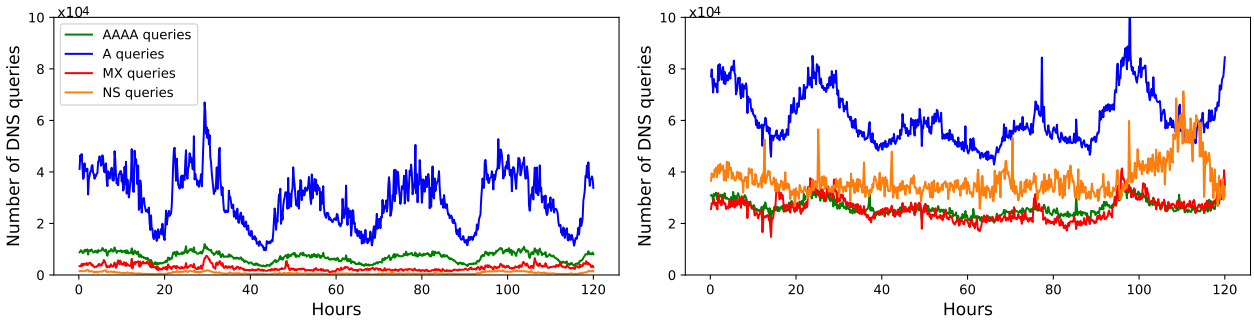
These two DNS servers were selected as they represent different scenarios for real authoritative TLD name servers, having different traffic statistics. As indicated in Table 2.1, the *Santiago* server received an average of 80 queries per second (QPS), whereas the *Amsterdam* server received an average of 263 QPS, that is, more than 3 times higher than the *Santiago* server. In addition, 25% of the DNS queries received by the *Amsterdam* server were queries

for nonexistent domains, in contrast with the *Santiago* server, where just 5% of the DNS queries were for nonexistent domains.

Table 2.1: Comparison of DNS traffic statistics between Santiago and Amsterdam datasets

	Santiago	**Amsterdam**
Average queries per second	80	263
% of DNS queries for nonexistent domains	5%	25%
% of DNS queries of type `A` (IPv4 address record)	66%	40%
% of DNS queries of type `AAAA` (IPv6 address record)	16%	17%
% of DNS queries of type `MX` (Mail exchanger record)	7%	14%
% of DNS queries of type `NS` (Name server record)	2%	22%

Figure 2.2 illustrates their differences in the number of DNS queries for four of the most requested record types: `AAAA`, `A`, `MX` and `NS`. The *Santiago* server mostly received DNS requests for `A` records (66%), and presents low percentages of requests for `MX` records (7%) and `NS` records (2%). On the other hand, the *Amsterdam* server received a lower percentage of requests for `A` records (40%), but higher percentages of requests for `MX` records (14%) and `NS` records (22%).



(a) Santiago dataset (b) Amsterdam dataset

Figure 2.2: Comparison of DNS traffic features between Santiago and Amsterdam datasets.

As it is well known, in the problem of anomaly detection in network traffic (and particularly in DNS traffic), the false-negative rate cannot be obtained since there is no prior certainty about the presence or absence of anomalies at each data point. Thus, the relevance of DNS anomaly detection methodologies cannot be evaluated using common measures such as recall.

Therefore, to test the AD-BoP method and compare it against other state-of-the-art methodologies, some artificially created anomalies were injected inside the DNS traffic of

both *Santiago* and *Amsterdam* datasets. Thereby, we generated a minimal set of data points that a DNS anomaly detector would be expected to detect. It is important to underline that this latter does not imply that the selection of these injected anomalies was influenced by our proposed method. Indeed, in the following, we provide an exhaustive explanation about why a DNS anomaly detector is expected to detect these traffic injections since they correspond to real-world anomalies that could (1) compromise the well-working of a TLD server or (2) use the normal operation of the DNS infrastructure as a weapon against a targeted server.

By using the tool mentioned in Section 2.4, 9 different DNS anomalies were injected into the last week of both datasets (last 1,008 `.pcap` files with 10-min traffic intervals). Each anomaly was injected inside a different 10-min traffic interval, and therefore, 9 out of the 1,008 `.pcap` files were labeled as certainly anomalous. In both *Santiago* and *Amsterdam* datasets, the 9 injected anomalies corresponded to 3 random subdomain attacks, 3 DNS amplification attacks, and 3 NXDOMAIN flood attacks. The injected attacks were varied in terms of magnitude, and they all correspond to DNS anomalies that an anomaly detection system should be able to detect, as they could (1) compromise the well-working of a TLD server (NXDOMAIN flood attacks) or (2) use the normal operation of the DNS infrastructure as a weapon against a targeted victim's server (DNS amplification and random subdomain attacks).

Table 2.2 illustrates the magnitude of the injected anomalies in both *Santiago* and *Amsterdam* datasets. In particular, each anomaly is labeled with its duration (ΔT) and with the average queries per second generated by the anomaly (QPS). Thus, each anomaly injected $\big(\text{QPS} \times \Delta\text{T}\big)$ DNS packets into the traffic received by the TLD servers.

Table 2.2: Magnitude of injected anomalies in both Santiago and Amsterdam datasets

	Santiago		Amsterdam	
	ΔT [s]	Avg. QPS	ΔT [s]	Avg. QPS
Random subdomain #1	30	100	30	100
Random subdomain #2	30	1,000	30	1,000
Random subdomain #3	30	10,000	30	10,000
DNS amplification #1	30	10	30	10
DNS amplification #2	60	10	60	10
DNS amplification #3	120	10	120	10
NXDOMAIN flood #1	30	100	30	600
NXDOMAIN flood #2	30	1,000	30	1,200
NXDOMAIN flood #3	30	10,000	30	2,400

As it can be seen, NXDOMAIN flood attacks increase the DNS traffic in at least one order of magnitude from the average queries per second normally received by both servers (Table 2.1). These magnitudes are consistent with the fact that NXDOMAIN flood attacks attempt to compromise the well-working of the TLD server receiving the queries (*Santiago* and *Amsterdam* servers). For random subdomain and DNS amplification, the TLD servers are not the target of the DoS attacks. Indeed, the real targeted victim's server could receive traffic from multiple TLD name servers simultaneously, and therefore, just a minor portion

of the entire attack's traffic will be visible inside *Santiago* and *Amsterdam* servers' traffic. Accordingly, the different magnitudes of these two types of DNS attacks are coherent with reality. On the one hand, random subdomain attacks cause DoS by generating many DNS queries to the targeted server. On the other hand, DNS amplification attacks cause DoS by generating large DNS responses (in bytes) to the targeted IP address. Thus, random subdomain attacks are more dependent on high QPS values than DNS amplification attacks, and therefore, DNS amplification anomalies were configured with lower QPS.

2.6 Experimental Results

This section presents the experimental results of testing the proposed DNS anomaly detection system (AD-BoP) in addition to the other two state-of-the-art methods (CZ.NIC method and QLAD-global). The three methodologies were tested on the last week of both *Santiago* and *Amsterdam* datasets, represented by 1,008 consecutive `.pcap` files, each one with 10 minutes of DNS traffic data. As mentioned before, both datasets contain nine artificially created anomalies among their 1,008 `.pcap` files, representing 0.9% of the 1,008 traffic windows. Thus, at each of these 1,008 time intervals, the methods are required to determine the presence or absence of anomalous traffic behavior.

As indicated in Sections 2.2 and 2.3, the three methodologies employ a sensitivity (or threshold) parameter to determine the presence of anomalies. Hence, an exhaustive threshold analysis was performed to evaluate the methods being compared at different configurations. In this analysis, several sensitivity values were tested, quantifying the total number of 10-min intervals labeled as anomalies (from 0 to 1,008) and the number of intervals with injected attacks correctly labeled as anomalies (from 0 to 9).

It is important to emphasize that the nine artificially injected DNS attacks corresponded to anomalies on a TLD server's normal operation that an anomaly detection system would be expected to detect. Thus, it is also relevant to identify the threshold values which allow the detection of all these nine attacks.

In the case of the AD-BoP method, it needed to create and train its LSTM model before analyzing the last week of both datasets. Thus, for each *Santiago* and *Amsterdam* server, AD-BoP first created a forecast model using the first three weeks of *Santiago* and *Amsterdam* datasets, respectively. These three weeks corresponded to the first 3,172 `.pcap` files of both datasets, preceding the 1,008 time intervals used for testing the anomaly detection methods. The 3,172 time intervals were split into train and validation sets using a 75% / 25% ratio to create LSTM forecast models according to the following configuration:

- **Training:** The model was fed with batches of size 24 and trained for 30 epochs using *Adam* optimizer and mean absolute error (MAE) as the loss function.
- **Hidden Layers:** An LSTM layer of size 150 follows with a dropout value of 0.3 and using hyperbolic tangent as the activation function.
- **Input:** The model receives as input one week of data, that is to say, 1,008 multivariate

values corresponding to 1,008 10-min traffic intervals.

- **Output:** The output of the model is a unique 9-dimensional value, which corresponds to the prediction of the DNS features for the next 10-min window. This value is used to retrain the model, updating its weight values before performing the next prediction.

With respect to the configuration of the CZ.NIC method, its default and recommended parameters were used: 8 aggregation levels, 25 hash functions, and hash tables of size 32.

2.6.1 *Santiago* dataset

Figure 2.3 shows the result of applying the AD-BoP methodology over the last week of the *Santiago* dataset using different threshold configurations. The red line shows the total number of 10-min windows labeled as anomalies (from 0 to 1,008) as a function of the threshold value. Similarly, the blue line shows the number of injected attacks correctly labeled as anomalies (from 0 to 9) as a function of the threshold value.

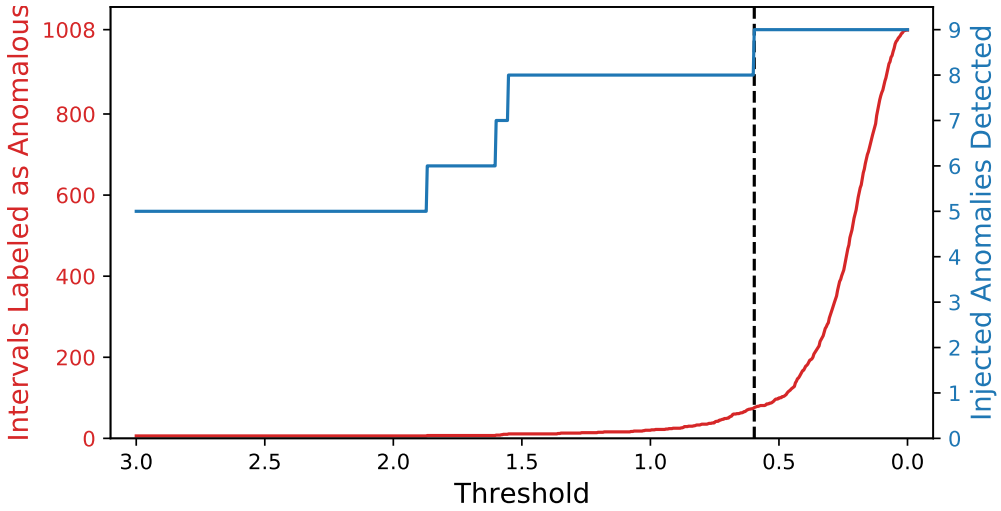


Figure 2.3: Anomaly detection in the *Santiago* dataset using AD-BoP method. In red: total anomalies detected in function of the threshold value. In blue: injected anomalies detected in function of the threshold value.

According to Figure 2.3, the threshold value needed to detect the nine injected anomalies is 0.684. This means that there is a difference of more than 68.4% between expected and real values for at least one traffic feature in the nine traffic windows with injected anomalies. At this threshold configuration, AD-BoP classified a total of 59 10-min windows as anomalous, out of a maximum of 1,008 traffic windows. That is, 5.9% of the whole *Santiago* dataset was labeled as anomalous behavior using this threshold.

When analyzing the number of total traffic windows labeled as anomalous in Figure 2.3 (red line), this curve is naturally separated into two different zones. Firstly, from a threshold value of 3.0 to approximately 0.5, there is a slow increase in the number of traffic windows labeled as anomalous. Secondly, from a threshold value of 0.5 to 0, there is a faster growth

of the curve, rapidly reaching the detection rate of 100% (all 1,008 intervals labeled as anomalous). With respect to the nine injected anomalies, they all were detected in the first slow-growing zone, being clearly separated from the vast majority of traffic intervals.

Figure 2.4 shows three of the nine DNS-related time series used by the AD-BoP method when analyzing the 1,008 10-min windows from the *Santiago* dataset. These time series correspond to 1) the number of distinct queried domains, 2) the number of DNS queries for `ANY` records, and 3) the number of DNS responses with response code `NXDOMAIN`. In addition, vertical red lines identify the position of the nine injected DNS attacks. This figure exemplifies one of the main claims on which this method is based: DNS anomalies can affect just a particular portion of the whole traffic. Hence, anomalies marked as 1, 5, and 7 correspond to DNS `NXDOMAIN` floods, which significantly impact the normal amount of DNS responses with code `NXDOMAIN` (Figure 2.4.c). Anomalies marked as 2, 3, and 8 correspond to DNS amplification attacks, and they significantly impact the normal traffic of DNS queries for `ANY` records (Figure 2.4.b). Lastly, anomalies marked as 4, 6, and 9 correspond to random subdomain attacks, significantly impacting the normal number of distinct requested domains (Figure 2.4.a).

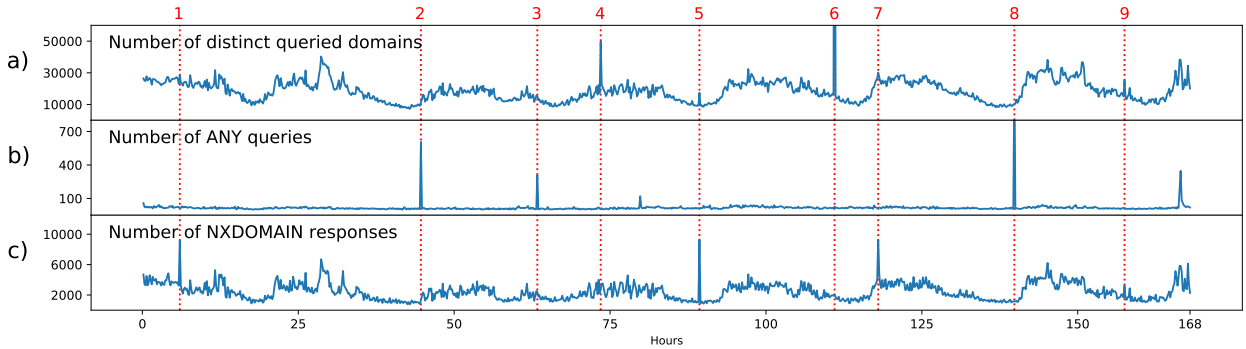


Figure 2.4: Time series for the last 1,008 10-min windows in the *Santiago* dataset, regarding a) the number of distinct queried domains, b) the number of `ANY` queries, and c) the number of responses for `NXDOMAIN`. In addition, the nine injected anomalies are identified with red vertical lines.

Furthermore, a visual inspection of the time series in Figure 2.4 reveals the abrupt changes in normal traffic behavior induced by the injected attacks. This supports the idea that DNS anomaly detection systems would be expected to detect these anomalies.

In addition, the CZ.NIC method was applied over the last week of the *Santiago* dataset using its two possible policies: source IP address policy and query name policy. As both policies use their own threshold configuration, they were analyzed separately, as shown in Figure 2.5. Using the source IP address policy, this method identified the nine injected anomalies at a threshold value of 1.199. At this configuration, the CZ.NIC method classified 975 10-min windows as anomalous, i.e., 96.7% of the whole dataset. On the other hand, the query name policy identified all injected anomalies with a threshold value of 1.779, classifying a total of 875 10-min windows as anomalous, i.e., 86.8% of all 10-min windows.

When analyzing the number of total traffic windows labeled as anomalous in Figure 2.5 (red lines), the curves for both policies present similar behaviors. The curves are naturally

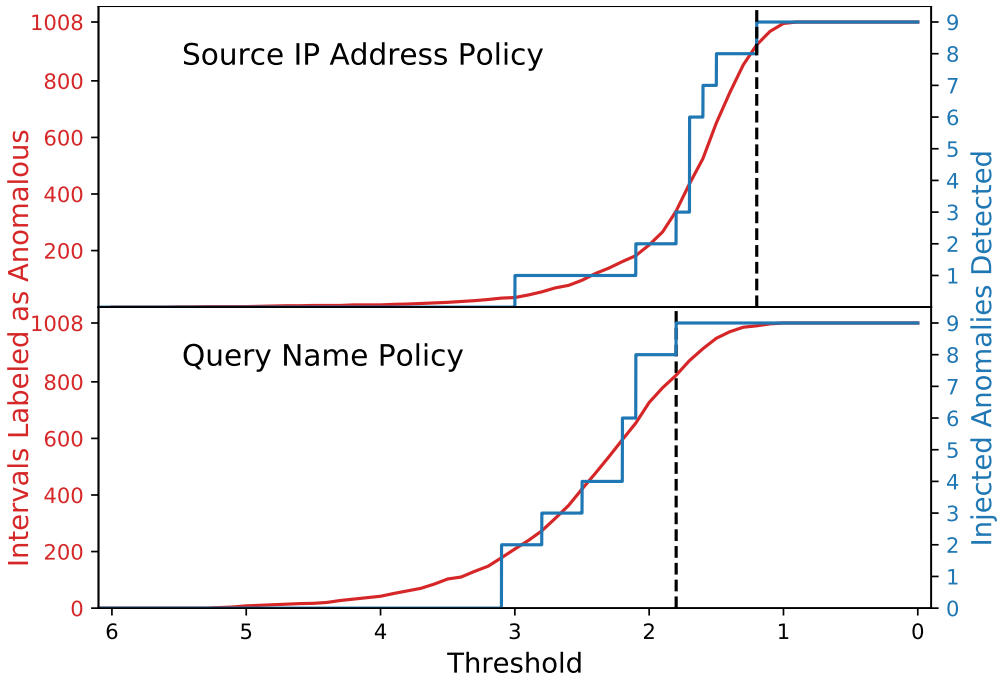


Figure 2.5: Anomaly detection in the *Santiago* dataset using the CZ.NIC method. The figure shows the behavior when using source IP address policy, and query name policy. In red: total anomalies detected in function of the threshold value. In blue: injected anomalies detected in function of the threshold value.

separated into two different zones: a first zone of slow growth, followed by a zone of faster growth. Differently from AD-BoP results, the nine injected anomalies were not detected in the first slow-growing zones. Moreover, the number of injected anomalies detected (blue lines) presents a similar behavior to the total number of intervals labeled as anomalous (red lines). Then, the detection of all these nine anomalies occurs at a high detection rate (more than 85% of all 1,008 intervals labeled as anomalous).

Since both policies were designed to complement each other, we also studied the behavior of the CZ.NIC method using both policies at the same time. Therefore, all the combinations of both thresholds that allow the detection of the 9 injected attacks were analyzed. The most efficient configuration to detect all these injected anomalies was found for both threshold values of 2.0. At this configuration, the CZ.NIC method classified a total of 779 10-min windows as anomalous, i.e., 77.3% of all traffic windows.

With respect to QLAD-global, Figure 2.6 shows the result of applying this methodology over the last week of the *Santiago* dataset using different threshold configurations. This method identified the nine injected anomalies at a threshold value of 1.224. At this sensitivity, QLAD-global labeled 548 10-min windows as anomalous. That is, 54.4% of all the 1,008 possible windows.

When analyzing the number of total traffic windows labeled as anomalous in Figure 2.6 (red line), the curve starts with a fast-growing behavior, rapidly reaching high detection rate values. This implies that, even when the detection of all the nine anomalies occurs rapidly, it appears with a high detection rate (more than 50% of all 1,008 intervals labeled

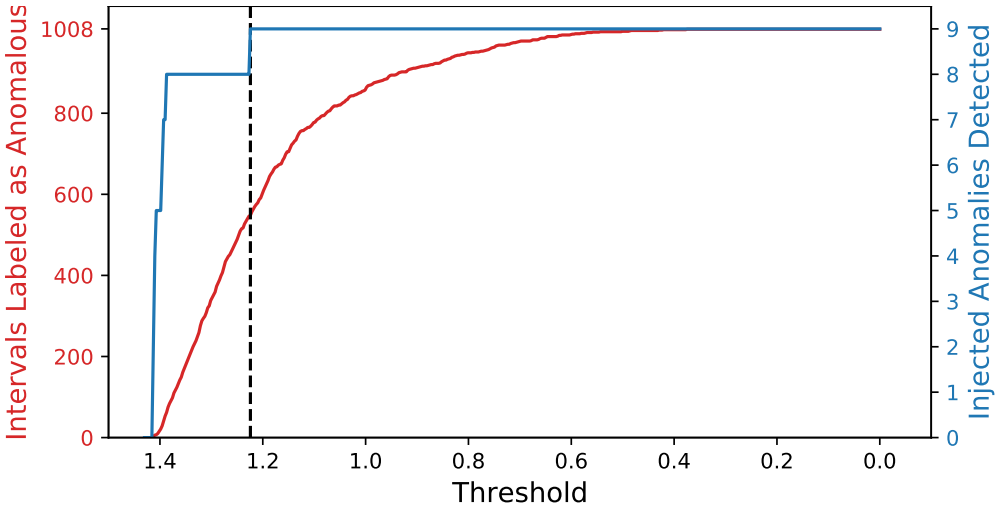


Figure 2.6: Anomaly detection in the *Santiago* dataset using QLAD-global. In red: total anomalies detected in function of the threshold value. In blue: injected anomalies detected in function of the threshold value.

as anomalous).

Finally, as QLAD-global was designed to complement the CZ.NIC method (referred to as QLAD-flow in [143]), we analyzed QLAD-global and the CZ.NIC method (using both policies) working together. Therefore, we analyzed all the combinations of thresholds that allow the detection of the nine injected attacks. The most efficient configuration to detect all these injected anomalies was found for QLAD-global's threshold of 1.387, query name policy's threshold of 2.4, and not considering source IP address policy. At this configuration, 511 traffic windows were classified as anomalous, i.e., 50.7% of the whole dataset.

2.6.2 *Amsterdam* dataset

For the *Amsterdam* dataset, the same experiments as for the *Santiago* dataset were performed.

Figure 2.7 shows the result of applying the AD-BoP methodology over the last week of the *Amsterdam* dataset using different threshold configurations. The threshold value needed to detect the nine injected anomalies is 0.348. This means that there is a difference of more than 34.8% between expected and real values for at least one traffic feature in the nine intervals with injected anomalies. At this threshold configuration, AD-BoP classified a total of 95 10-min windows as anomalous, out of a maximum of 1,008 traffic windows. That is, 9.4% of the whole *Amsterdam* dataset was labeled as anomalous by using this threshold.

When analyzing the number of total traffic windows labeled as anomalous in Figure 2.7 (red line), this curve is naturally separated into two different zones, in the same way as for the *Santiago* dataset. Firstly, from a threshold value of 3.0 to approximately 0.2, there is a slow increase in the number of traffic windows labeled as anomalous. Secondly, from a threshold value of 0.2 to 0, there is a faster growth of the curve, rapidly reaching the detection rate

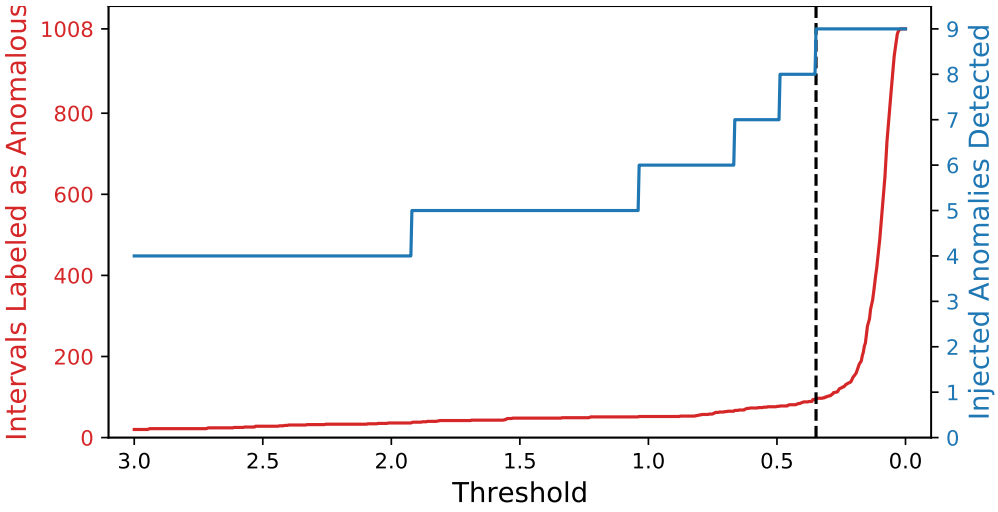


Figure 2.7: Anomaly detection in the *Amsterdam* dataset using AD-BoP method. In red: total anomalies detected in function of the threshold value. In blue: injected anomalies detected in function of the threshold value.

of 100% (all 1,008 intervals labeled as anomalous). As for the *Santiago* dataset, the nine injected anomalies were all detected in the first slow-growing zone, being clearly separated from the vast majority of traffic intervals.

Similarly to the *Santiago* dataset, Figure 2.8 shows three of the nine DNS-related time series used by the AD-BoP method when analyzing the 1,008 traffic windows from the *Amsterdam* dataset. These time series correspond to 1) the number of distinct queried domains, 2) the number of DNS queries for `ANY` records, and 3) the number of DNS responses with response code `NXDOMAIN`. In addition, vertical red lines identify the position of the nine injected DNS attacks. As for the *Santiago* case, this figure exemplifies the fact that DNS anomalies can affect just a particular segment of the whole traffic. Anomalies marked as 1, 5, and 7 correspond to DNS `NXDOMAIN` floods, which significantly impact the normal amount of DNS responses with code `NXDOMAIN` (Figure 2.8.c). Anomalies marked as 2, 3, and 8 correspond to

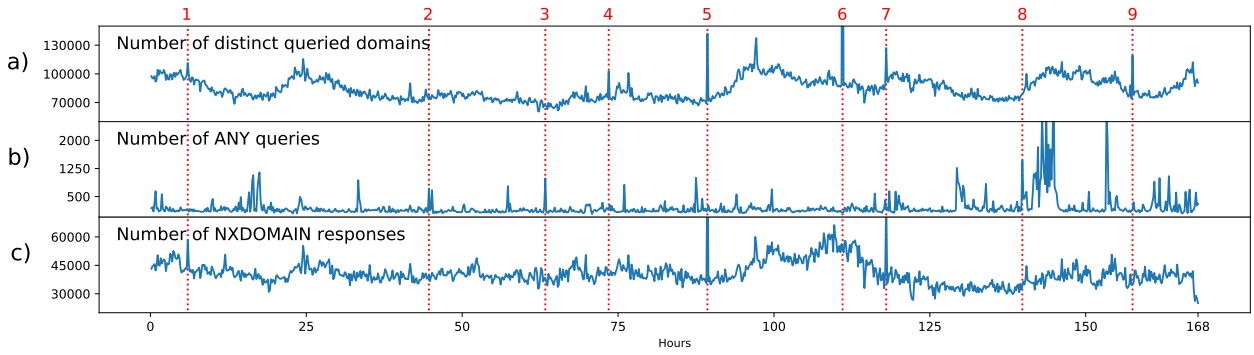


Figure 2.8: Time series for the last 1,008 10-min windows in the *Amsterdam* dataset, regarding a) the number of distinct queried domains, b) the number of `ANY` queries, and c) the number of responses for `NXDOMAIN`. In addition, the nine injected anomalies are identified with red vertical lines.

DNS amplification attacks, and they significantly impact the normal traffic of DNS queries for `ANY` records (Figure 2.8.b). Lastly, anomalies marked as 4, 6, and 9 correspond to random subdomain attacks, which significantly impact the normal amount of distinct requested domains (Figure 2.8.a).

Regarding the CZ.NIC method, it was applied over the last week of the *Amsterdam* dataset using its two possible policies with different threshold configurations, as shown in Figure 2.9. Using the source IP address policy, this method identified the nine injected anomalies at a threshold value of 1.399. At this configuration, the CZ.NIC method classified 1,006 10-min windows as anomalous, i.e., 99.8% of the whole dataset. On the other hand, the query name policy identified all injected anomalies with a threshold value of 0.699, classifying a total of 1,008 10-min windows as anomalous, i.e., 100% of all 10-min windows.

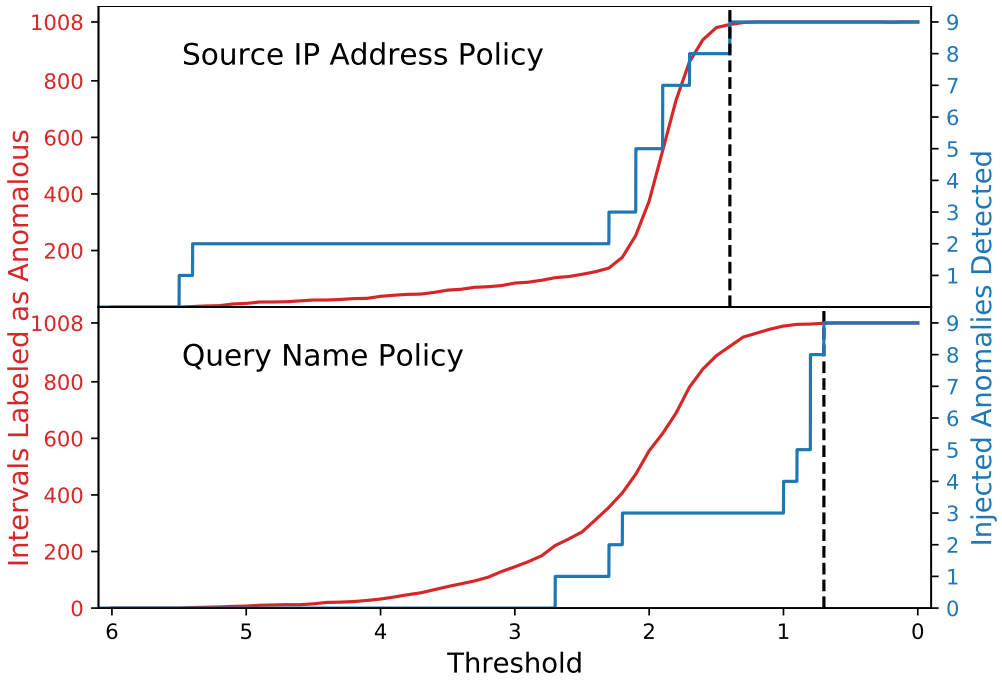


Figure 2.9: Anomaly detection in the *Amsterdam* dataset using the CZ.NIC method. The figure shows the behavior when using source IP address policy, and query name policy. In red: total anomalies detected in function of the threshold value. In blue: injected anomalies detected in function of the threshold value.

When analyzing the number of total traffic windows labeled as anomalous in Figure 2.9 (red lines), the curves for both policies present similar behaviors. As for the *Santiago* dataset, the curves are naturally separated into two different zones: a first zone of slow growth, followed by a zone of faster growth. The nine injected anomalies were not detected in the first slow-growing zones, replicating the behavior obtained in the *Santiago* dataset. Then, the detection of all these nine anomalies occurs at a high detection rate (more than 90% of all 1,008 intervals labeled as anomalous).

In addition, when using both policies simultaneously, it was found that the most efficient configuration to detect all injected anomalies was to only consider the source IP address policy.

With respect to QLAD-global, Figure 2.10 shows the result of applying this methodology over the last week of the *Amsterdam* dataset using different threshold configurations. This method identified the nine injected anomalies at a threshold value of 1.161. At this sensitivity, QLAD-global labeled 773 10-min windows as anomalous. That is, 76.7% of all the 1,008 possible traffic windows.

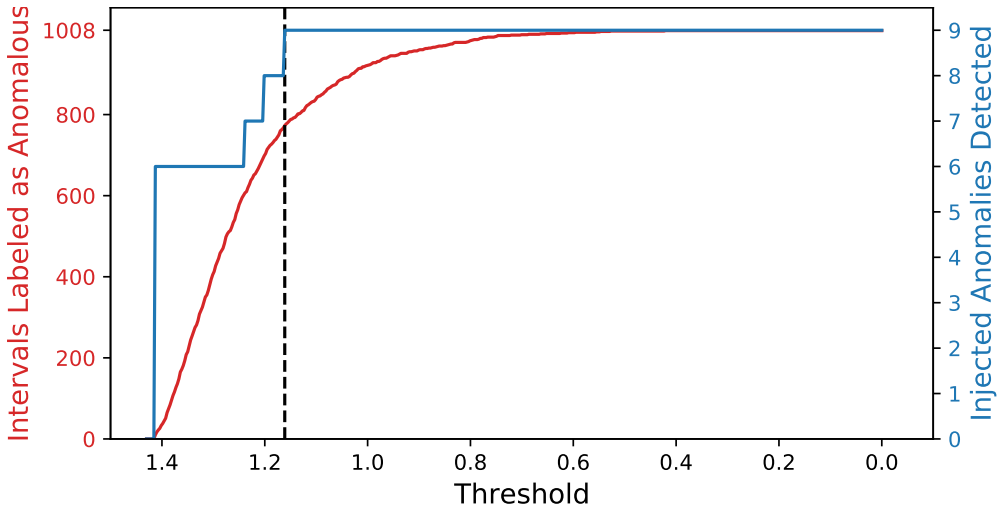


Figure 2.10: Anomaly detection in the *Amsterdam* dataset using QLAD-global method. In red: total anomalies detected in function of the threshold value. In blue: injected anomalies detected in function of the threshold value.

When analyzing the number of total traffic windows labeled as anomalous in Figure 2.10 (red line), the curve starts with a fast-growing behavior, rapidly reaching high detection rate values. As for the *Santiago* dataset, this implies that, even when the detection of all the nine anomalies occurs rapidly, it appears with a high detection rate (more than 75% of all 1,008 intervals labeled as anomalous).

Finally, we also analyzed QLAD-global and the CZ.NIC method (using its two policies) together. Thereby, the most efficient configuration to detect all the injected anomalies was found for QLAD-global's threshold of 1.409, source IP address policy's threshold of 5.3, and query name policy's threshold of 2.6. At this configuration, 262 traffic windows were classified as anomalous, i.e., 26.0% of the whole dataset.

Table 2.3 summarizes, for each of the compared methodologies, the percentage of 10-min windows labeled as anomalous when detecting the nine injected anomalies.

2.7 Discussion

2.7.1 Evaluation of Detected Anomalies

As mentioned in Section 2.6, the performance of the anomaly detection methodologies was analyzed based on the detection of a minimal set of artificially created DNS attacks. These

Table 2.3: Percentage of traffic windows labeled as anomalous at the first detection of the nine injected anomalies.

	Santiago	**Amsterdam**
CZ.NIC source IP policy	96.7%	99.8%
CZ.NIC query name policy	86.8%	99.9%
CZ.NIC both policies	77.3%	99.8%
QLAD-global	54.4%	77.7%
QLAD-global + CZ.NIC both policies	50.7%	26.0%
AD-BoP	**5.9%**	**9.4%**

nine injected attacks corresponded to anomalies that a successful DNS anomaly detector would be expected to detect. Consequently, we studied the threshold values that allow the methods to detect all the nine injected anomalies. Thus, we quantified the total number of traffic windows labeled as anomalous when detecting all these DNS attacks. This quantification can be used to measure the quality of each method, since if a method requires a very low threshold to detect the injected anomalies, it will classify a large number of normal traffic windows as anomalous, increasing the number of false positives.

According to Table 2.3, the CZ.NIC method labeled in all cases more than 75% of the whole dataset as anomalous when detecting the nine DNS attacks, even when applying both policies simultaneously. Similarly, QLAD-global also presents a high percentage of 10-min windows labeled as anomalies: 54.4% for the *Santiago* dataset and 77.7% for the *Amsterdam* dataset.

It is important to notice that these large numbers of traffic windows classified as anomalies are not incorrect per se, since there are some cases where anomalous traffic can be actually more frequent than normal traffic [45], e.g., when dealing with bursty DDoS attacks. However, when analyzing both *Santiago* and *Amsterdam* datasets with DNS experts from NIC Chile, we did not find situations similar to that case. Therefore, the total number of real anomalies should not be as high as for QLAD-global and the CZ.NIC methods. This claim is well aligned with the reasonable and widely accepted assumption that the majority of the network connections are normal traffic, vastly outnumbering the number of anomalous traffic [130].

Joining QLAD-global with the CZ.NIC method using its two policies (as proposed by Robberechts et al. [143]) enhanced the effectiveness of detecting the injected DNS attacks. That is, the number of total traffic windows labeled as anomalies was reduced to 50.7% for the *Santiago* dataset and to 26.0% for the *Amsterdam* dataset. This decrease is especially important for the *Amsterdam* dataset, as when applied separately, all the methods classified more than 75% of the whole dataset as anomalous. Nonetheless, this approach still incurs a high number of traffic intervals labeled as anomalies for the *Santiago* dataset (50.7% of the whole dataset). Furthermore, this ensemble method is based on the proper selection of three different thresholds (as discussed in Section 2.6), which can be a complicated task to be properly achieved in practice.

Different from the other approaches, the AD-BoP method presented a lower percentage

of 10-min windows labeled as anomalies when detecting the nine injected attacks: 5.9% for the *Santiago* dataset and 9.4% for the *Amsterdam* dataset. These results are more coherent with the widely accepted assumption of a low percentage of anomalous network traffic [130].

According to these results, AD-BoP classified as anomalous 59 traffic intervals from the *Santiago* dataset, where only nine corresponded to the injected anomalies. After inspecting the remaining 50 detected anomalies along with DNS experts from NIC Chile, it can be concluded that:

- 14 detected anomalies corresponded to heavy spamming/email marketing. These traffic anomalies came mostly from 10 different IP addresses, where 4 of them were listed in spam blocklists.
- 21 detected anomalies corresponded to DNS enumeration activities, including standard record enumeration (`A`, `NS`, `SOA`, and `MX` records) and `SRV` enumeration. These traffic anomalies came mostly from 14 different IP addresses, where 10 of them were found in IP blocklists.
- 9 detected anomalies corresponded to the query behavior of caching resolvers.
- 2 detected anomalies corresponded to random subdomain attacks. The IP address performing these anomalies was found in at least 5 different IP blocklists.
- 1 detected anomaly corresponded to a DNS amplification attack using spoofed queries of type `ANY`.
- The remaining 4 detected anomalies could not be clearly identified with any real anomalous behavior and were considered as false positives.

Therefore, when detecting the nine injected anomalies, AD-BoP incurred a false-positive rate of 6.8%.

In the same way, AD-BoP classified as anomalous 95 traffic intervals from the *Amsterdam* dataset, where only nine corresponded to the injected anomalies. After inspecting the remaining 86 detected anomalies along with DNS experts from NIC Chile, it can be concluded that:

- 56 detected anomalies corresponded to subdomain enumeration using `ANY` records. The presence of these anomalies is clearly visible in Figure 2.8.b, where the number of `ANY` queries is highly affected by these anomalies, especially at the end of the time series.
- 4 detected anomalies corresponded to heavy spamming/email marketing, with a highly distributed pattern.
- 12 detected anomalies corresponded to DNS enumeration activities using standard record enumeration (`A`, `NS`, `SOA`, and `MX` records). These traffic anomalies came mostly from 22 different IP addresses, where 10 of them were found in IP blocklists.

- 7 detected anomalies corresponded to DNS amplification attacks using spoofed queries of type `ANY`.
- The remaining 7 detected anomalies could not be clearly identified with any real anomalous behavior and were considered as false positives.

Therefore, when detecting the nine injected anomalies, AD-BoP incurred a false-positive rate of 7.4%.

When analyzing the results obtained by AD-BoP in both datasets, it can be shown that this method is able to detect a wide range of other real DNS anomalies besides the artificially created ones, incurring false-positive rates lower than 8%.

2.7.2 Threshold Evaluation

As the three methods being compared employ a threshold value to adjust their detection sensitivities, an exhaustive threshold analysis was performed in Section 2.6 to evaluate the implications of different threshold values.

Considering that these methods are intended to be used by TLD registry operators, they will need to understand the meaning of the threshold value. A proper interpretation of the threshold value could be crucial in selecting a proper sensitivity configuration for their services.

Regarding the AD-BoP method, its threshold value λ can be explained directly: it denotes the maximum accepted difference between the expected and real values of any DNS feature t with reference to the expected value, i.e.,

$$\frac{|y_t - y'_t|}{y'_t} > \lambda \tag{2.8}$$

This easily explainable threshold value leads to easy interpretability of detected anomalies. For example, if the threshold is set to 0.5, a detected anomaly can be explained as a difference of more than 50% between the real and the expected value for one of the DNS-related features (number of queries for `A` records, number of distinct requested domains, number of queries for nonexistent domains, etc.).

With respect to the CZ.NIC method, the interpretation of its threshold value is more complicated since it does not have a simple relation with the DNS traffic being analyzed. This is because its threshold is applied after a complex data processing of the DNS data, comprehending the use of different hash functions, gamma distribution fitting, and the use of Mahalanobis distance, among other tasks. Consequently, configuring the sensitivity of the CZ.NIC method is difficult since there is no clear relationship between a given threshold value and the expected behavior of the anomaly detection system.

In the case of QLAD-global, its threshold value can be roughly interpreted as a maximum accepted difference between the expected and real entropy values of the selected DNS features.

As the threshold refers to the normalized entropy of the DNS features being analyzed, there is no preconception about the impact of possible anomalies on normalized entropies' values. Therefore, it is complicated to select a meaningful threshold value to be used in real scenarios. Also, when analyzing Figure 2.6 and Figure 2.10, the number of total detected anomalies rapidly increases when selecting threshold values lower than 1.4. This behavior makes it difficult to adjust a proper threshold value since very small variations can lead to the inclusion of a large number of false positives.

2.7.3 Performance Evaluation

In this subsection, the performance in terms of execution time is analyzed for the three anomaly detection methods being compared. Accordingly, the average time needed to label a 10-min traffic window as normal or anomalous was reported for each method. In the following, all values refer to the use of a 1.6GHz quad-core processor (Intel Core i5-8250U) with 8GB RAM.

The CZ.NIC method required the shortest time to process a single 10-min `.pcap` file and classify it as normal or anomalous. When using its source IP address policy, CZ.NIC method required an average time of 0.24 s for the *Santiago* dataset and 0.36 s for the *Amsterdam dataset*. On the other hand, when using its query name policy, it required an average time of 0.94 s for the *Santiago* dataset and 1.53 s for the *Amsterdam* dataset.

Regarding QLAD-global, it required an average time of 10.32 s for the *Santiago* dataset and 36.84 s for the *Amsterdam* dataset. These values are considerably higher than the presented by Robberechts et al. [143], as QLAD's authors explicitly did not count the time needed to extract the traffic feature distributions from the `.pcap` files. However, we did count this time as the extraction of features from log files cannot be performed *offline*, and it must be completed before deciding the presence or absence of an anomaly.

With respect to the proposed AD-BoP method, a more detailed analysis is performed. When AD-BoP receives a new `.pcap` file for a time interval t, it performs as follows:

i) It analyzes the `.pcap` file to extract a vector $\hat{y}_t$ with the value of the nine DNS-related features for interval t.

ii) It compares these real values in vector $\hat{y}_t$ with the vector of predicted values $\hat{y}'_t$ (calculated in the previous interval $t-1$). According to the value of threshold λ, it decides if the interval t presents an anomaly or not.

iii) It updates its LSTM network model with real values in vector $\hat{y}_t$.

iv) It performs a one-step-ahead prediction and obtains a vector $\hat{y}'_{t+1}$ with the nine predicted traffic features for the next interval $t+1$.

Accordingly, to perform anomaly detection on a given 10-min traffic interval, only steps i and ii are necessary. Steps iii and iv must be completed before the next time interval, but they are not needed to decide the presence or absence of anomalies in the current interval.

Hence, to label a 10-min window (steps i and ii), AD-BoP required an average time of 7.9 s for the *Santiago* dataset and 28.3 s for the *Amsterdam* dataset. Steps iii and iv (which are not needed to label the current time window) were performed in an average time of 3.8 s for both *Santiago* and *Amsterdam* datasets.

Table 2.4 summarizes the average time needed by each method to classify a 10-min traffic interval as normal or anomalous. Thereby, the AD-BoP method is shown to be competitive to the other methodologies in terms of execution time, considering its outstanding performance in correctly detecting the injected anomalies.

Table 2.4: Average time needed to decide the presence or absence of an anomaly for a 10-min traffic window

	Santiago	**Amsterdam**
CZ.NIC source IP policy	0.24 s	0.36 s
CZ.NIC query name policy	0.94 s	1.53 s
QLAD-global	10.32 s	36.84 s
AD-BoP	**7.90 s**	**28.30 s**

2.7.4 Time Series Forecasting Method

As mentioned in Section 2.3, the fundamental basis for the well-working of AD-BoP is its prediction step. To this end, we selected an LSTM model to forecast the DNS-related time series. This selection was based on its suitability for predicting DNS traffic, capturing the periodic patterns in the traffic, and detecting some abrupt phase changes [94]. However, the prediction step in AD-BoP methodology can be carried out by using other time series techniques. For example, methods more light-weighted than LSTM networks can be considered. In the following, we present a comparison of AD-BoP's results by using the LSTM model and Holt-Winters [31] as time series forecasting methods. Both methods are compared in terms of their effectiveness in detecting the injected anomalies and their execution time.

As in Section 2.7.1, the performance of both versions of the AD-BoP method was studied based on the detection of the nine injected attacks. Therefore, we analyzed the total number of traffic windows labeled as anomalous when detecting all the artificial DNS attacks. According to Table 2.5, for both *Santiago* and *Amsterdam* datasets, AD-BoP was less efficient when using Holt-Winters as the forecasting method.

Table 2.5: Percentage of traffic windows labeled as anomalous at the first detection of the nine injected anomalies. Comparison of both versions of the AD-BoP method.

	Santiago	**Amsterdam**
AD-BoP LSTM	5.9%	9.4%
AD-BoP Holt-Winters	58.4%	24.8%

In addition, we studied the performance of both versions of AD-BoP according to their

execution time. As mentioned in Section 2.7.3, when AD-BoP receives a new `.pcap` file, it performs four steps. In the first two steps, AD-BoP extracts the features from the raw file and compares these real features against the values predicted in the previous time interval. These two initial steps are independent of the forecasting method being used, and therefore, the average time needed to decide the presence or absence of an anomaly is the same for both versions of AD-BoP: 7.90 s for the *Santiago* dataset and 28.30 s for the *Amsterdam* dataset, as stated in Table 2.4. In contrast, the last two steps are different for each prediction method. In the third step, AD-BoP updates (or refits) the prediction model considering the `.pcap` file just arrived. In the fourth step, the forecasting method performs a one-step ahead prediction of the nine traffic features for the next interval. According to Table 2.6, Holt-Winters models were faster on average to be updated and perform prediction.

Table 2.6: Average time needed to update (or refit) the prediction model and perform a one-step ahead prediction.

	Santiago and **Amsterdam**
AD-BoP LSTM	3.8 s
AD-BoP Holt-Winters	2.52 s

Nevertheless, as mentioned in Section 2.7.3, these two final steps are not needed to label the current traffic window as normal or anomalous. These procedures must be completed before the arrival of the next `.pcap` file, and therefore, their execution time must be bounded by 10 minutes. However, as shown in Table 2.4, both methods required much less than 10 minutes to complete steps iii and iv. Consequently, the improvement in execution time obtained by Holt-Winters does not have real significance.

These results are an example of a well-known trade-off between prediction goodness and complexity. In our particular case, the reduction in false positives (when using LSTM) is much more relevant than the reduction in the time needed to predict new values (when using Holt-Winters). In addition, as discussed in Sections 2.7.1, 2.7.2, and 2.7.3, AD-BoP (using LSTM) improves the current state-of-the-art for DNS anomaly detection in authoritative TLD name servers. Therefore, using an LSTM model is shown to be appropriate and fit the requirements of the DNS anomaly detection problem.

2.7.5 Limitations of the current work

Differently from the two state-of-the-art methodologies used for comparison (CZ.NIC method and QLAD-global), our proposed AD-BoP method requires an initial training process to create the LSTM forecast model. Some works point to this training methodology as something to be avoided for DNS anomaly detection due to the need for a large amount of historical data to feed the machine learning models [143]. However, we do not consider this a real problem for TLD registry operators, as they can collect all the data directly from the normal operation of their own name servers.

A major concern about this study is that, as we lack labeled data, we could be training our LSTM forecast model using DNS traffic with some anomalies inside. We can then im-

plicitly make the model learn these anomalies and use them to define normal traffic behavior. This is, in fact, not only a consideration for our AD-BoP method but a common concern for general unsupervised methodologies. Because of this concern, unsupervised network anomaly detection techniques are based on the assumption that only a small percentage of traffic is malicious [44, 1], and that the model learned during training is robust to these few anomalies [46]. However, we claim that learning the normal DNS traffic behavior by using data with some anomalies inside is not necessarily something wrong. This is exemplified in Figure 2.11, which shows the number of DNS queries for `A` records received by one of NIC Chile's authoritative name servers during a period of three weeks. In this real example, there is a repetitive anomaly with a clear 24-hour periodicity. As this anomaly presents a highly repetitive pattern, it can actually become part of the TLD server's expected traffic. In that case, this anomaly will no longer meet the consensual high-level definition of an anomaly, referring to a pattern that does not conform to what is expected [30, 45, 2]. Thus, it is debatable whether the recurrent anomaly eventually became part of normal traffic, considering that a machine learning technique could actually expect its occurrence.

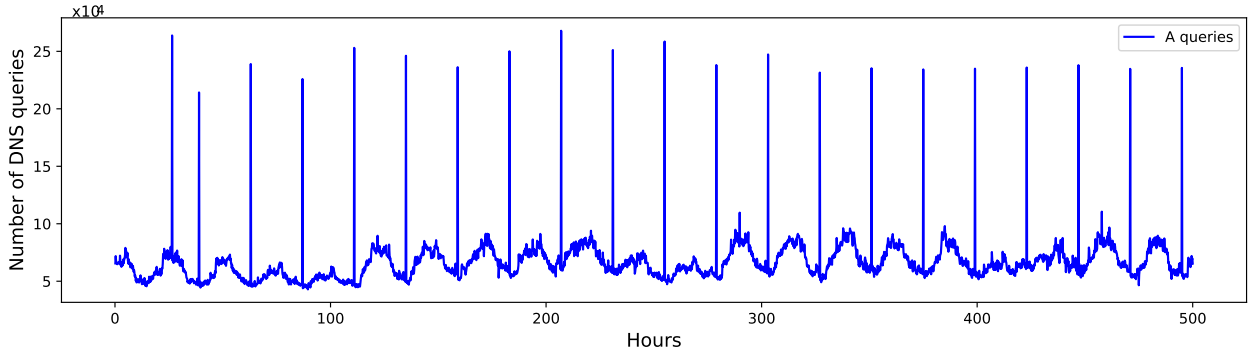


Figure 2.11: Number of DNS queries for `A` records received by one of NIC Chile's authoritative name servers during 3 weeks.

2.8 Summary

In this chapter, we presented a methodology for Anomaly Detection Based on Prediction (AD-BoP), by using a machine learning model to forecast different portions of the whole DNS traffic. AD-BoP provides a useful and easily explainable methodology to effectively detect DNS anomalies, which could be especially helpful for TLD registry operators to preserve the reliability of their services.

After an exhaustive analysis, the proposed method was demonstrated to improve the state-of-the-art methodologies for DNS anomaly detection in TLD name servers. AD-BoP outperformed these methodologies by efficiently detecting a set of artificially created DNS anomalies, presenting low false positive rates and near real-time execution times.

Different from other previous works, this study was performed using large amounts of DNS traffic, considering an entire month of normal operation from two authoritative TLD name servers. Thus, this work and its results take on great importance and relevance.

To enhance the readability of our findings, AD-BoP used the same threshold configuration for all the traffic features for detecting anomalies. However, a TLD operator may be interested in assigning different sensitivity configurations to each traffic feature. Therefore, the effectiveness of AD-BoP can be outperformed by selecting different threshold values for each feature.

Chapter 3

Network Measurements from Mobile Devices

3.1 Introduction

As of 2023, almost 60% of all web traffic comes from mobile devices [159], and moreover, these values are expected to increase over the next years. Thus, almost any Internet-related study will likely pay great attention to mobile users. Indeed, mobile users' devices are a key target for deploying network measurement systems, as they can directly relate to Internet traffic, network performance, mobile applications usage, and mobile network coverage, among others.

Guided by the main objectives of this thesis, we scrutinized different network measurements that could be taken directly by mobile devices and that could be used for anticipatory networking purposes. Therefore, in order to perform this study, we put efforts into developing a network measurement tool capable of taking a broad range of key performance indicators closely related to mobile users' QoS.

In this chapter, we present a novel measurement methodology for mobile devices composed of *i)* passive monitoring of all Internet traffic, and *ii)* a comprehensive contextualization of user context, including channel quality metrics and observed quality of service in Internet connections. Our approach is able to monitor network traffic without the need for user intervention or root permissions to run properly, by taking advantage of Android's APIs and Linux kernel facilities. Thus, we can periodically run our system to gather statistical information about all established UDP and TCP connections, including average throughput and passive ping-like measurements from all TCP connections, such as RTT, jitter, and the number of lost packets. Moreover, our presented measurement methodology includes a contextualization phase where the monitored traffic information is enriched with a comprehensive set of environmental information. The dataset collected by our monitoring tool, served as a basis to the proposed user prediction methodology, which is further described in Chapter 4.

3.2 Background

3.2.1 Multidimensional QoS Analyses

State-of-the-art works on characterizing mobile QoS usually perform multidimensional analysis to describe network performance. Several authors [120, 59, 23, 101] looked at end-to-end performance by taking active measurements in user-space and correlating multiple QoS indicators with user's contextual information. Our work differs from theirs as we perform passive measurements on real user traffic. In many works [145, 164, 14, 78], QoS indicators of TCP flows were obtained from inside mobile carriers using passive monitoring probes. Thus, they were not able to collect user's contextual information and, moreover, the estimated QoS indicators may not match what the user perceived. Indeed, no comprehensive work in the related literature was dedicated to measuring the QoS of all existing Internet connections in mobile devices by only taking passive measurements in user-space.

3.2.2 Network Ping Indicators

Among all factors that affect the network service quality, Round-trip time, jitter, and packet loss are three QoS indicators that have been thoroughly studied due to their great influence on the overall network's performance [79]. Both academic and engineering researchers have analyzed these QoS indicators for network management purposes, such as their use in TCP stacks to help optimize bandwidth usage [160].

In addition, RTT, jitter, and packet loss indicators have received a great deal of attention due to their huge impact on the user's quality of experience (QoE). Unfavorable network conditions may translate into high values of these indicators, which can be perceived by the user in the form of network delays or inability to establish connections. Their impact in QoE has already been studied in some scenarios such as content delivery networks [26], multiplayer gaming [15], HTTP video streaming [108], VoIP [70] and popular mobile services and apps [28].

The most common tool for measuring RTT, jitter, and packet loss is the Ping networking tool. The Ping utility sends ICMP (Internet Control Message Protocol) packets of type `echo request` and awaits for ICMP packets of type `echo reply`. There are two main drawbacks when using this tool for measuring a device's received QoS. i) It does not provide precise information on the received QoS because traffic control mechanisms have different rules for ICMP and TCP/UDP traffic [160, 171] and ii) as most active measurements, Ping only measures the quality of a single connection and not the whole state of the user's Internet connectivity.

3.2.3 Passive Ping Approaches

As a way to face these problems, researchers and network operators have adopted passive Ping approaches, enabling the analysis of every data flow without inducing a traffic overload

that could affect their measurements. The following methods are some of the known passive approaches for estimating TCP RTT for a device's Internet connections: i) capturing the timestamp of TCP handshake packets [67], ii) timing TCP packets in a remote measurement machine [100], iii) analyzing the `timestamp` option in the TCP header [87] following the reflection mechanism described by Jacobson et al. [65], which allows to any capture point on the connection's path to measure the round trip delay, and iv) capturing the timestamp of TCP handshake packets in a remote monitoring probe [145]. Unlike these approaches, the periodic passive Ping method presented in this work (PePa Ping) is able to obtain RTT, jitter and the number of lost packets from within the device's system by taking advantage of the Linux kernel's TCP facilities. Our method does not need any user intervention, root permission or remote measurement machine, making it easy to distribute to real users.

3.2.4 Traffic Monitoring Through Local VPN

Even though there are many projects related to measuring QoS on mobile networks, to the authors' knowledge, this is the first to use a local VPN approach. We call local or internal VPN to a mounted or simulated VPN server inside the client device itself, acting like a man-in-the-middle proxy. Different from our work, most of the projects that monitor mobile networks using an internal VPN are focused on malware and privacy leakage detection [154, 141]. PrivacyGuard [154] is a VPN-based Android platform that detects information leakage by intercepting applications' network traffic using a local VPN. While it uses the same approach as this work, PrivacyGuard needs to be able to read all traffic information in plain text even if it comes encrypted using TLS protocol. This means doing TLS interception and packet inspection. Our measurement approach does not incorporate any overhead of such type due to its passive nature. Our work only uses the VPN to retransmit the packets through its own sockets, which allows us to easily measure RTT, jitter, and the number of lost packets. On the same line, there is Haystack [141], a VPN-based Android platform that analyzes mobile Internet traffic, locally and in user-space, in order to characterize mobile traffic and detect privacy risks. Our work uses the same user-space VPN approach as Haystack, with the main difference being that i) we implemented its sockets in C++, which allows obtaining QoS-related metrics for each flow directly from the Linux kernel, and ii) it runs with a periodic behavior, showing a lower impact on the device's battery in comparison to a continuous monitoring system.

3.3 Measurement Methodology

As the core of our measurement methodology, we developed PePa Ping (*Periodic Passive Ping*), an Android framework capable of taking periodic *Ping-like* measurements (RTT, jitter, and the number of lost packets) through passive monitoring of active connections without injecting any new packet into the network, contrary to standard Ping and its ICMP packets. The above is achieved by taking advantage of Android WorkManager API, Android VpnService, and querying the OS directly for QoS information.

In the following subsections, we will explain the methodology behind PePa Ping, which

is represented by its three main characteristics: being periodic, passive, and *Ping-like*.

3.3.1 Periodic Behavior

One of the main concerns when developing a monitoring system for mobile devices is the overhead it may produce in CPU and battery consumption. This is the reason for choosing a periodic monitoring system over a continuous one. With enough periodicity, we can obtain a good representation of the network's traffic without overloading the device.

WorkManager is an API provided by Android since 2018. This API makes it easy to schedule deferrable, asynchronous tasks that are expected to run even if the app exits or the device restarts [9]. By using this API, we schedule our monitoring system to start running for 1 minute every 15 minutes. We chose 15-minute intervals because it is the minimum duration that WorkManager allows for periodic tasks to be scheduled without them interfering with the device's battery optimization system. Regarding the monitoring time of our system, we chose 1-minute slots as an attempt to impact the normal operation of users' devices minimally. Nevertheless, given the regularity of both human mobility patterns [122] and Internet usage patterns [175], the selected methodology is expected to be representative of the whole Internet traffic generated by users.

3.3.2 Passive Monitoring

According to RFC 7799 [110], passive methods have three main characteristics: they are i) based solely on observations of an undisturbed and unmodified packet stream of interest, ii) dependent on the existence of one or more packet streams to supply the stream of interest, and iii) dependent on the presence of the packet stream of interest at one or more designated Observation Points. While trying our best to follow these properties, we designed a system that uses Android's VpnService to implement a local VPN server written in C++. Next, we will explain this in detail.

VpnService

PePa Ping implements an Android VpnService in order to gain packet-level access without requiring root privileges. According to Android's documentation, VpnService is a base class for applications to extend and build their own VPN solutions [8]. VpnService gives a file descriptor `fd` to the application for it to read and write on. Each read from `fd` retrieves an outgoing IP packet which was routed to the interface. Each write to `fd` injects an incoming IP packet just like it was received from the interface. A typical VPN-client application completes the VPN connection by processing and exchanging packets with the remote server over a tunnel. In our case, we complete the VPN connection by handing the packets over to our local VPN server, which handles the traffic processing.

Local VPN Server (Packet Forwarder)

Our local VPN server is implemented in C++, and it is able to run within the application's Java code by using the Java Native Interface (JNI) framework. The local server receives the outgoing IP packets through the VPN's `fd` and sends only its payload, without any headers, through an application-level socket of type `SOCK_DGRAM` for UDP or type `SOCK_STREAM` for TCP. Given that the VPN's `fd` only receives IP packets, but reading from an application-level socket returns a TCP/UDP packet's payload, it is necessary to manually handle its headers before injecting them into the `fd`. Creating artificial headers differs for UDP and TCP. In both cases, we have got to calculate the IP checksum. The custom UDP header is simple because of its connectionless nature, and the main work consists of calculating the packet length and the UDP checksum. TCP is more complex because it involves keeping track of handshake packets and acknowledgment and sequence numbers of data packets. This procedure is explained in further detail in the work of Razaghpanah et al. [141].

The local VPN keeps track of the current connections using two C++ unordered maps. These maps store TCP and UDP connections represented by objects of a C++ class called `VpnConnection`, with the map's unique-key being the concatenation of the source port, destination IP, and destination port. The `VpnConnection` object stores the socket that communicates our local VPN with the destination address. Furthermore, we implement a polling method using the system call `epoll`, used to monitor multiple file descriptors and generate an event signaling that I/O is possible on any of them. Having a polling method is necessary for high-performance networking applications with non-blocking socket I/O.

Even though other authors have implemented similar local VPN approaches [141, 154], they implemented the packet forwarding mechanism in Java. We chose to implement it in C++ because it grants us access to specific connection-level information, as mentioned in the following section.

3.3.3 Ping Information

Our system registers the opening and closing time of every TCP and UDP socket using the POSIX function `gettimeofday()` that provides current time with microsecond (μs) precision. In addition, when a TCP socket (`SOCK_STREAM` type) is closed, we call `getsockopt()` function with the option `TCP_INFO` that retrieves a `struct tcp_info` variable. This variable gives us access to information about the closed TCP connection, directly from the Linux kernel. Using this method, we gain access to every TCP connection's `rtt`, `rtt_var` and `lost_packets` variables, similar to the information obtained by the Ping tool. The `rtt` and `rtt_var` variables are estimations of RTT mean and RTT standard deviation (commonly referred to as jitter), which are used for TCP congestion control [131, 20]. The Linux kernel calculates these estimations using Jacobson's algorithm [10, 64].

It is important to clarify that even when the network performance perceived by the users may be affected by the overhead of our local VPN server, there's no overhead in `rtt`, `rtt_var` or `lost_packets` measurements, as they are taken from the sockets connected directly from our local VPN server to the destination address. Therefore, these measurements characterize

the real QoS of the established connections.

As Android is based on a modified version of the Linux kernel, it would be possible to obtain these low-level quality indicators using other client-side monitoring approaches such as *Wireshark* and *tcpdump*. However, these methods require to cross-compile the *libpcap* library for Android in order to include it inside the users' devices. Moreover, *libpcac* must be run with superuser privileges to work properly, and therefore, these methods can only be executed in rooted Android phones. Consequently, these approaches are not suitable for a crowdsourcing scenario.

3.4 Mobile Crowdsourcing App

To test our measurement methodology with real users, we built an Android application that incorporates the PePa Ping framework described in Section 3.3. As mentioned in Section 3.3.2, we collected information from all TCP and UDP connections throughout our system runs (PePa Ping methodology). By using our C++ VPN approach, we collected the following information for all network flows: destination IP address and port, network protocol, start time and end time (with microsecond precision), the number of bytes transmitted and received, and package name of the Android application that established the connection. In addition, for all TCP connections, we collected the following information by accessing the Linux kernel `tcp_info` structure: Average round-trip time (RTT), minimum RTT, RTT variance (jitter), and the number of lost packets.

Additionally, our Android application continuously collected contextual information during each 1-min execution of PePa Ping. This information allows for a better understanding of the conditions in which the Internet connections of a specific 1-min execution were generated. The contextual information of each 1-min execution can be divided into five groups: *User*, *Connectivity*, *Cellular*, *WiFi*, and *Ram*.

- *User*: Contains the information that remains invariable throughout the whole minute of measurement: start timestamp of the 1-min execution, an identifier of the user device, and an identifier of the network operator that issued the device's SIM card.
- *Connectivity*: Contains information related to Internet connectivity changes between Wi-Fi and Mobile Data, i.e., timestamp of each event, type of Internet connection (Wi-Fi or Mobile Data), and the number of the autonomous system to which the sending router belongs. The connectivity changes are tracked by registering an Android broadcast receiver for the `android.net.conn.CONNECTIVITY_CHANGE` intent.
- *Cellular*: Contains information related to the cellular network quality throughout the 1-min execution of the measurement system. It is important to note that the information collected varies according to the mobile network technology being used. We used the Android `PhoneStateListener` class for monitoring changes regarding the observed cell and for monitoring changes regarding the network signal quality. Each measurement includes a timestamp of the event, information about the cell type (by

inspecting the corresponding Android `CellInfo` object), information about the network technology being used (using the `getNetworkType()` function), and the identifier of the antenna used to access the mobile network. Additionally, there are different signal quality indicators being measured depending on the mobile network technology, such as the Received Signal Strength Indicator (RSSI), the Channel Quality Indicator (CQI), the Reference Signal Received Power (RSRP), the Reference Signal Received Quality (RSRQ), the Reference Signal Signal-To-Noise Ratio (RSSNR), the Received Signal Code Power (RSCP), the bit error rate (BER), among others. It is important to mention that each execution of the PePa Ping methodology can include a set of *cellular* events even if the device has not been using the mobile network to access the Internet (for example, if using Wi-Fi). The above is because, in those cases, the cell phone is still connected to the mobile network.

- *WiFi*: Contains information about the Wi-Fi quality throughout the 1-min execution of the measurement system. Each measurement includes the timestamp of the event, the Received Signal Strength Indicator, and the Wi-Fi frequency. These changes are tracked by registering a broadcast receiver for the `WifiManager.RSSI_CHANGED_ACTION` Android intent.

- *RAM*: Contains information about the RAM usage throughout the 1-min execution of the measurement system. Each RAM measurement includes the timestamp of the event, the total amount of RAM, and the available free RAM. The changes in the RAM usage are tracked by parsing the `/proc/meminfo` virtual file every 5 seconds.

Figure 3.1 shows an example of our measurement methodology. The figure shows different applications establishing Internet connections using the mobile network. Some of these connections experienced a vertical handover between the mobile network technologies LTE and WCDMA. It is important to notice that we can precisely determine which connections were affected by the handover events and which were not.

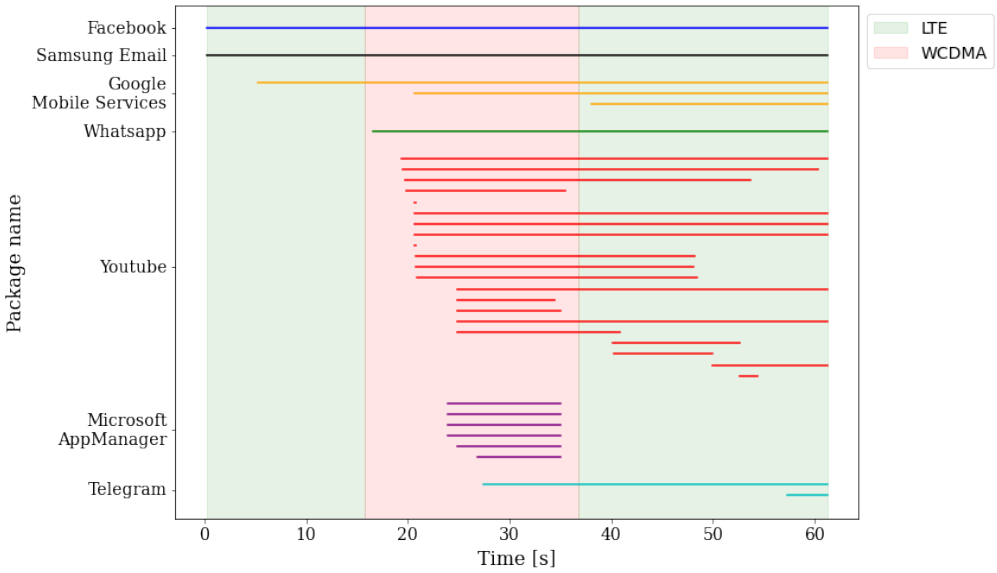


Figure 3.1: Example execution of our methodology with the presence of vertical handovers.

Figure 3.2 presents an example of the information collected by our measurement application in a 1-min execution. The figure shows different Android applications that establish connections of different durations throughout the 60 seconds of measurement. In addition, a variety of contextual information is also displayed as a set of time series accompanying the connections (signal quality information and % of used RAM). These time series provide a comprehensive contextualization of the 1-min execution, showing all the temporal changes in detail. Moreover, these time series provide a precise contextualization of every single connection established.

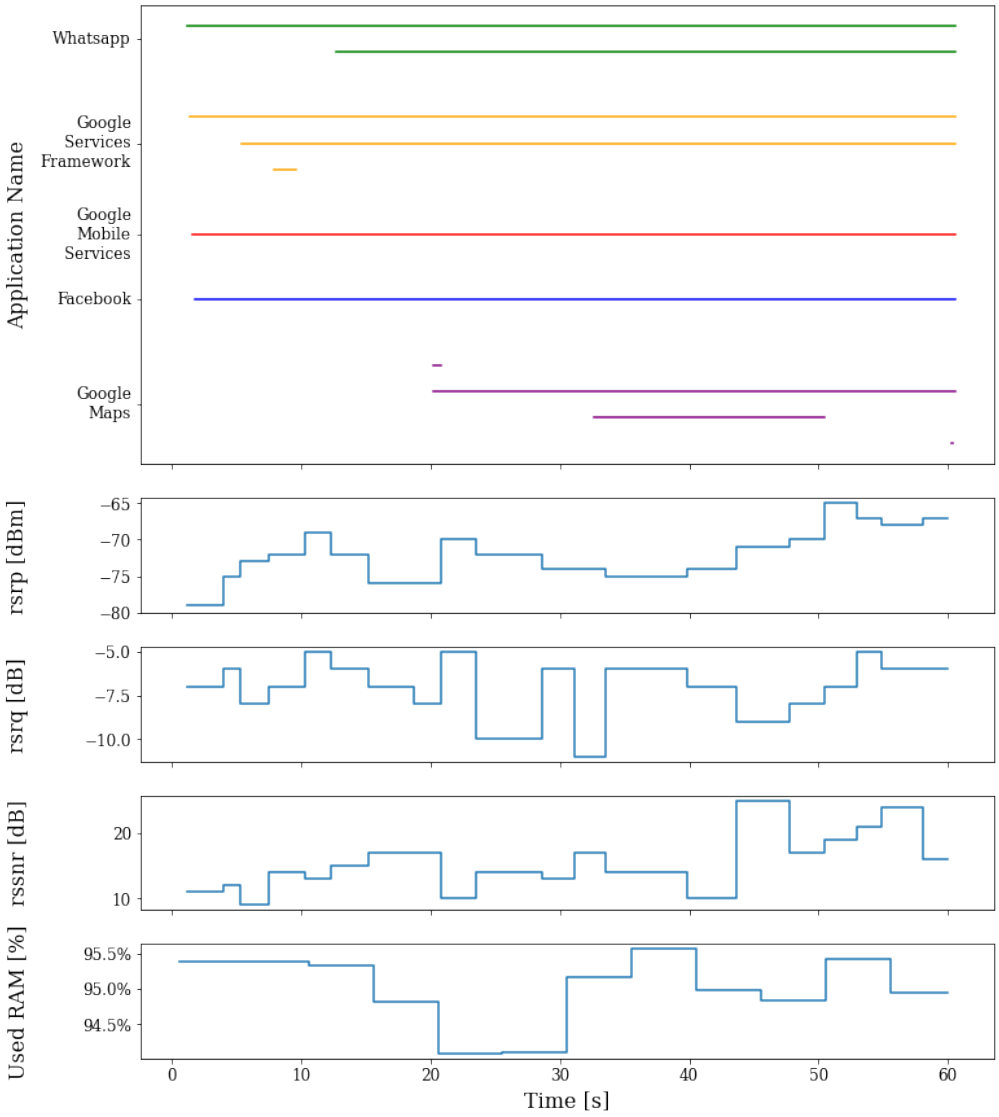


Figure 3.2: Sample of the information collected during an execution of our methodology.

In order to distribute the application to real users, we also put our efforts into creating a user interface that gave useful information about the received quality of mobile Internet. We implemented the following features: i) daily information about overall latency and throughput, ii) latency and throughput comparison with the rest of the users, and iii) daily information about latency and throughput with respect to some popular applications/ser-

vices. Figure 3.3 shows the main view of our Android application and the implemented features.

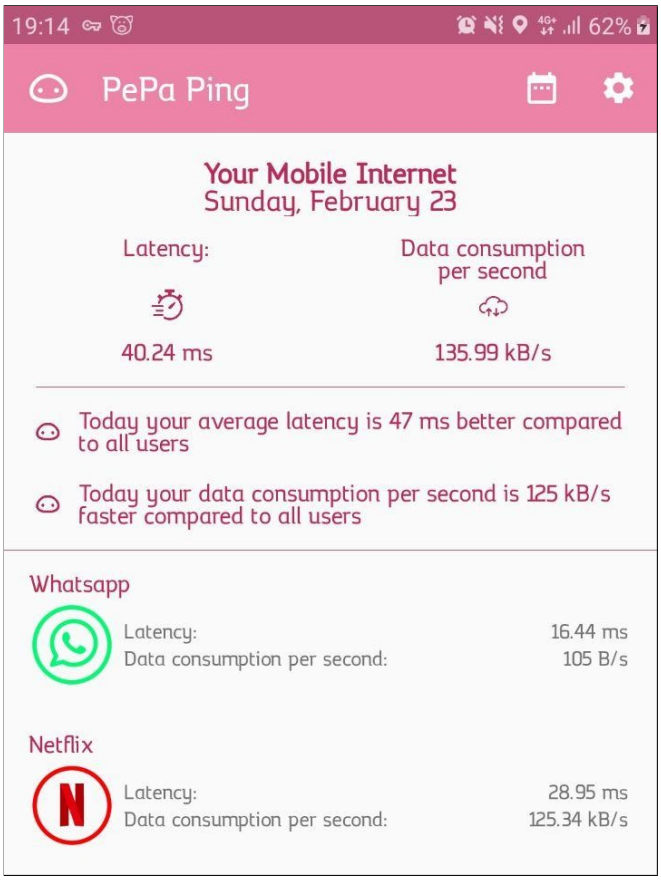


Figure 3.3: Main view of PePa Ping Android application

Additionally, we developed a database server to store all collected data in a PostgreSQL database. User devices sent their data to this server through secure HTTP connections (HTTPS).

3.4.1 Introduced Battery Overhead

To measure the introduced overhead on battery consumption by our PePa Ping system, we used Android `dumpsys` tool [6] specifying the `batterystats` service to gain access to power use information (in mAh) per UID (application) and system component from the command line. Our experiment consisted of performing a 30-second network-intensive task (access to a predetermined streaming content) with and without the PePa Ping system running in background. In both cases, we registered the energy consumption of:

1. System components (e.g. screen, WiFi, and cell)
2. System apps (Android apps permanently running in background)

3. The network-intensive app

4. PePa Ping app

Both experiment configurations were repeated 10 times, and no other user's apps were running beside the network-intensive app and the PePa Ping app. Table 3.1 compares the average energy consumption by each device's component while running the experiment. As expected, system components do not present a significant impact due to the execution of PePa Ping. In contrast, there is a clear decrease in the power use of system apps and the network-intensive app when PePa Ping is enabled. This is explained as these apps are no longer in charge of accessing the network by themselves, as this work is done by the PePa Ping system. Thus, the energy consumed when accessing the network is attributed to the PePa Ping app and not to the apps that are really requesting content from the network.

Accordingly, an interesting finding is that even though PePa Ping adds a new source of energy consumption, its impact on the total power consumption is very low ($< 2\%$). As explained before, PePa Ping frees other applications from consuming energy to access the Internet, as its local VPN server manages all Internet connections. This is evidenced in the higher decrease in energy consumption exhibited by the network-intensive app in comparison to the system apps. Thus, the impact on the battery life of real users should be even less significant, as our PePa Ping system is not continuously running in background in contrast to the presented experiment.

Table 3.1: Network-intensive experiment: Introduced overhead on battery consumption

Item	**Power Use without PePa Ping [mAh]**	**Power Use with PePa Ping [mAh]**	**PePa Ping Impact**
System Components	5.458	5.449	−0.16%
System Apps	6.824e−2	3.199e−2	−53.12%
Network-intensive App	4.235e−1	1.314e−2	−96.90%
PePa Ping App	0	5.676e−1	–
TOTAL	**5.95**	**6.062**	**1.88%**

3.5 Data Collection

Between 2020 and 2021, we conducted two different recruitment campaigns, where we distributed our crowdsourcing application to students from the Faculty of Physical and Mathematical Sciences of the University of Chile, who voluntarily accepted installing the application on their personal devices. A description of both datasets is provided below:

2020 Dataset:

- Number of real users: 160
- Time period: 3 months
- Number of executions of the 1-min measurement system: ∼240,000
- Number of monitored Internet traffic flows: ∼3,050,000
- Number of monitored TCP flows: ∼2,700,000
- Number of monitored UDP flows: ∼350,000
- Number of distinct destination IP addresses: ∼40,000
- Number of monitored Android applications: 831

2021 Dataset:

- Number of real users: 137
- Time period: 2 months
- Number of executions of the 1-min measurement system: ∼220,000
- Number of monitored Internet traffic flows: ∼3,200,000
- Number of monitored TCP flows: ∼2,300,000
- Number of monitored UDP flows: ∼900,000
- Number of distinct destination IP addresses: ∼30,000
- Number of monitored Android applications: 1255

A more technical explanation of the collected data can be found in the project's repository [75].

Ethical Considerations. In order to keep the ethical considerations of network measurements and management of user data, this study was conducted ensuring that full permissions are sought from and provided by the users of the measurement app [76]. After being shown a full description of the PePa Ping project (the collected information, how data will be used, and who will access and use the dataset), we asked participants to grant us permission to use their collected information for project-related research activities before installing the crowdsourcing application. Collected measurements in the presented dataset were appropriately anonymized to ensure the privacy of participants.

3.6 Inspection of the collected dataset

The main purpose of inspecting the collected dataset was to inquire about relationships among different metrics, in order to bring forward a methodology for predicting throughput in mobile networks. However, the presented measurement methodology and the collected dataset have been shown to be useful for a broad range of analyses, including but not limited to the analysis of different Internet trends (such as protocol adoption and application usage) and the analysis of network performance (such as ISP comparison and network coverage). Some of these correlated insights are presented in Appendix A.

As exemplified by Figure 3.1 and Figure 3.2, one of the main novelties of our passive measurement methodology is the comprehensive contextualization of each individual network flow, including a detailed description of signal quality. More precisely, the collected dataset allows us to properly compute the average signal quality associated with each network flow individually. As it is well known that channel quality impacts the performance of communications, we can employ the collected data to study the problem of mobile QoS prediction based on signal quality. Thus, we can correlate the performance metrics associated with each network flow (such as RTT, jitter, and average throughput) to signal quality metrics (such as RSSI and SINR).

Based on rudimental results in information theory, network performance is often modeled as a function of channel quality. However, the collected dataset exhibit that attempting to find a one-to-one correspondence between mobile QoS and signal quality is a limited approach, because several sources of variability will be ignored. In the following, we describe some variables that impact the performance of the TCP connections observed by our passive measurement tool. It is important to note that even when our main focus is to study mobile user throughput, these preliminary analyses are based on other network performance indicators (RTT and jitter). Nevertheless, these two QoS metrics are strongly related to throughput in cellular networks [158].

Destination of Traffic

PePa Ping stores the destination IP of each traffic flow, and therefore, each connection destination can be georeferenced to a particular continent. Latency values are likely to present differences regarding their geographic destination, as RTT values are determined in part by the physical distance and the infrastructure between both ends. In fact, Figure 3.4 presents the different patterns of RTT and jitter values for the four destination continents with the higher number of connections: North America, South America, Europe, and Asia. The figure clearly evidences the impact of the geographic destination when studying mobile QoS.

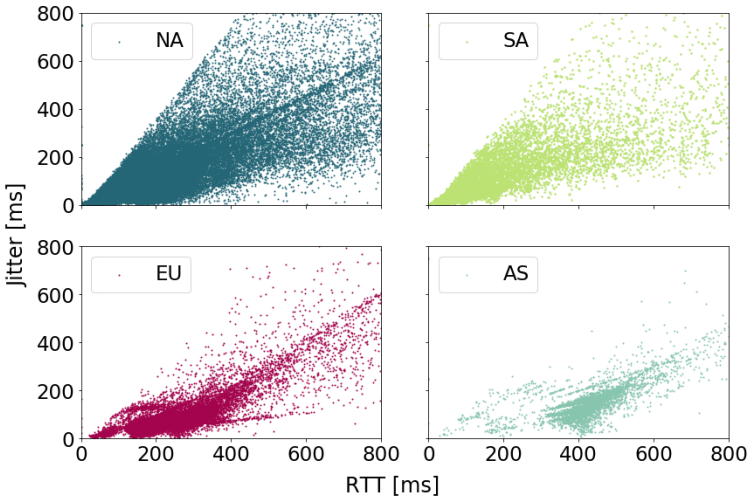


Figure 3.4: RTT and jitter for all TCP connections in mobile networks to different continents.

Regarding the distribution of RTT on different continental destinations, Figure 3.5 shows that TCP connections to South America and North America presented a similar behavior. Indeed, both regions showed higher density on lower RTT values, in contrast to the higher RTT levels for Europe and Asia.

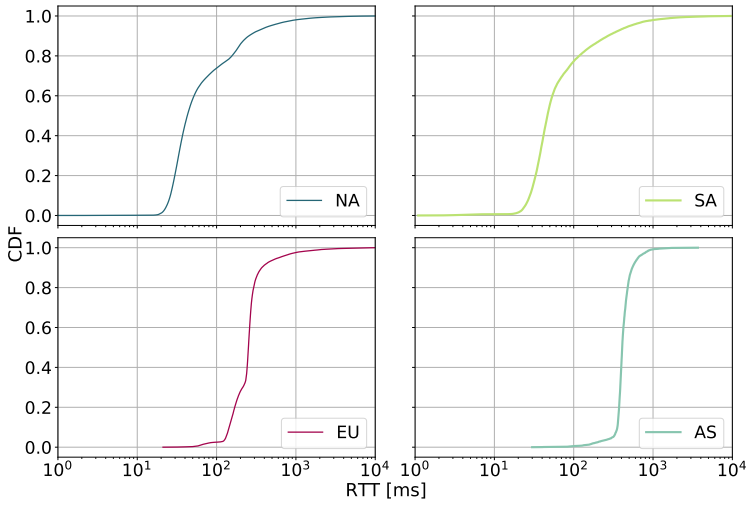


Figure 3.5: Cumulative distribution function (CDF) of RTT for all TCP connections in mobile networks among different continental destinations.

With respect to the distribution of jitter values on different continental destinations, Figure 3.6 illustrates that jitter distributions presented similar patterns to the RTT distributions. Indeed, TCP connections to South America and North America presented higher density on lower jitter values, in contrast to the higher jitter levels for Europe and Asia.

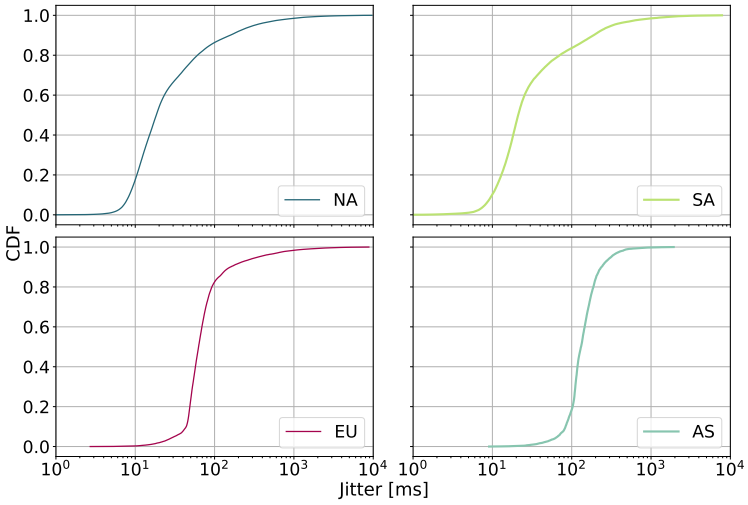


Figure 3.6: Cumulative distribution function (CDF) of jitter for all TCP connections in mobile networks among different continental destinations.

According to the previous Figures 3.5 and 3.6, South America and North America presented the lowest RTT and jitter values. As the measurements were collected from users in Chile (a South American country), it is reasonable that connections to South America obtained lower RTT and jitter values with respect to the connections to farther geographic locations. In addition, the fact that connections to North America obtained RTT and jitter distributions very close to the obtained for South America can be explained by the physical links between Chile and North America. At the present time, North America (and particularly the United States) is the only geographic location outside South America that is connected directly to Chile using submarine fiber optic cables [163].

Network Technology

As different network technologies intrinsically present different physical capacities, they are expected to impact the QoS of TCP connections in terms of RTT and jitter. Figure 3.7 shows the impact of the mobile network technology and the received signal strength on the RTT values of TCP connections. There is a tendency for all network types to increase the measured RTT values at lower signal strength, which confirms previous works [120]. Therefore, to maintain consistency, it is always important to analyze both signal quality and network type together, as the impact of signal quality on the overall mobile QoS depends on the mobile network technology being used, as shown in Figure 3.7.

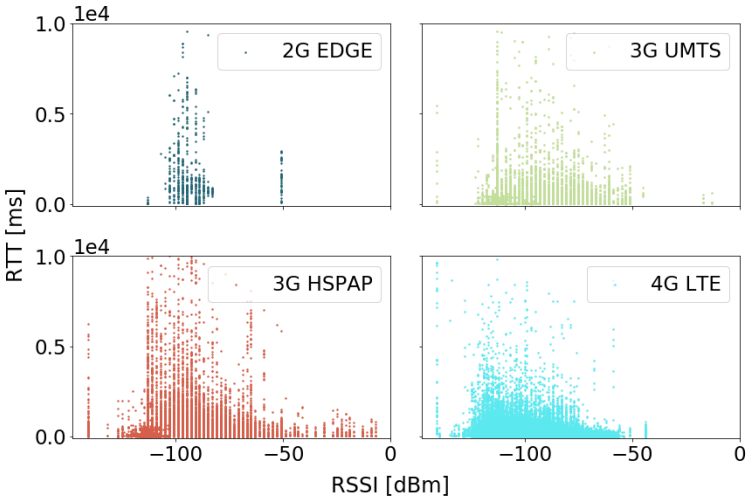


Figure 3.7: Impact on RTT of mobile network type and RSSI

Mobile Network Operator

The differences between their physical infrastructures and their traffic policies contribute to mobile operators behaving differently from each other. Indeed, the high variability in terms of throughput and latency across different mobile network operators has already been evidenced in previous studies [120]. Figure 3.8 illustrates the differences in terms of RTT and jitter values obtained by the three major mobile network carriers in Chile. Therefore, according to the data collected by PePa Ping, RTT and jitter values appear to be also influenced by the mobile carrier.

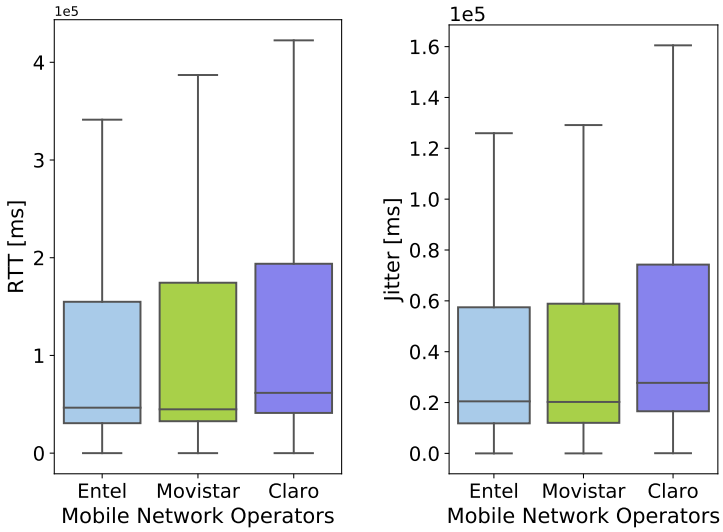


Figure 3.8: Impact on RTT and jitter of different mobile network operators.

It is important to mention that the values presented in Figure 3.8 are coherent with OpenSignal's reports for Chile [98], where *Entel* is stated as the Chilean mobile network

carrier presenting the best mobile experience.

Handovers

Despite the contextual variables previously reviewed, there are other external variables that, even though they do impact the measured RTT and jitter values, they are unknown at the start time of every connection. This situation restricts the limits of predictability in mobile QoS, since there is some crucial information that is not available to characterize the levels of RTT and jitter in mobile connections.

Some of these unknown parameters that have an impact on the mobile network performance are external to the network, such as weather conditions [93, 89] and crowd movements [41, 152]. Nevertheless, there are also internal events that impact the mobile network performance and that cannot be foreseen, such as the occurrence of handovers.

Previous works have discussed the negative impact of handovers on the performance of mobile networks [145, 49], and the sudden increase of RTT values after a handover [50]. As our monitoring system continuously obtains network contextual information, we can use cell identifiers to detect the occurrence of handover events during the execution of our monitoring system. Moreover, we can clearly distinguish between horizontal handovers (between the same network technology) and vertical handovers (between different network technologies).

Figure 3.9 shows the impact on the values of RTT in the presence of vertical and horizontal handovers when the connections are established via different network technologies. Likewise, Figure 3.10 presents the impact on the values of jitter in the presence of vertical and horizontal handovers when the connections are established via different network technologies.

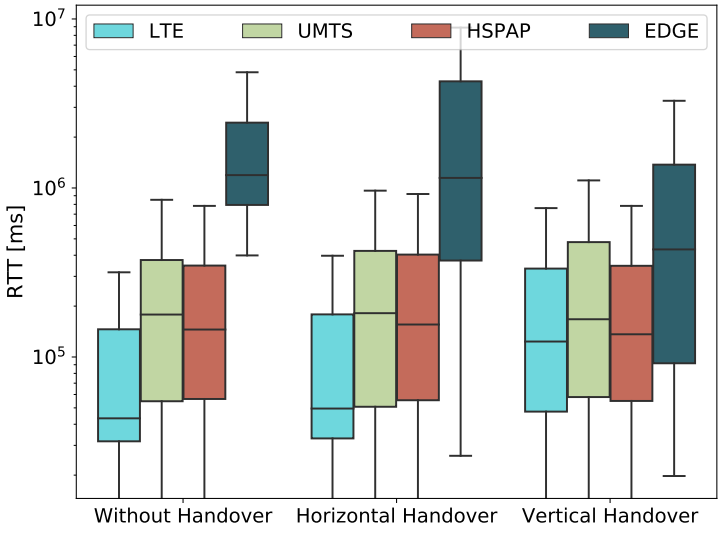


Figure 3.9: Impact on RTT of cell handover among different network technologies

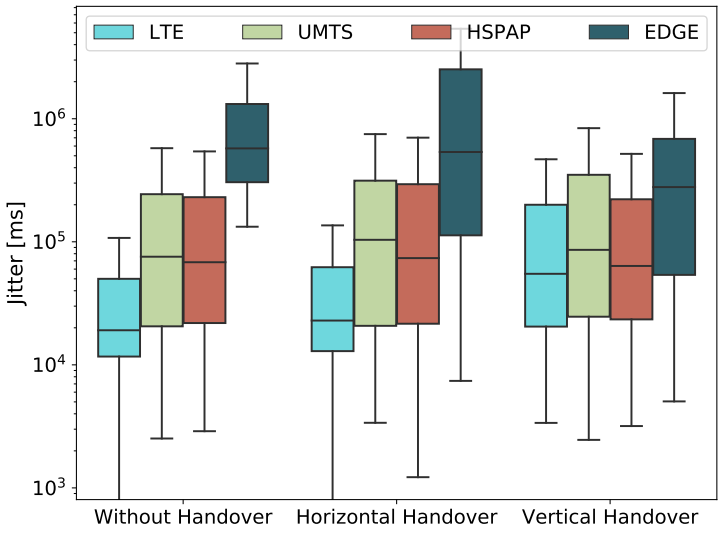


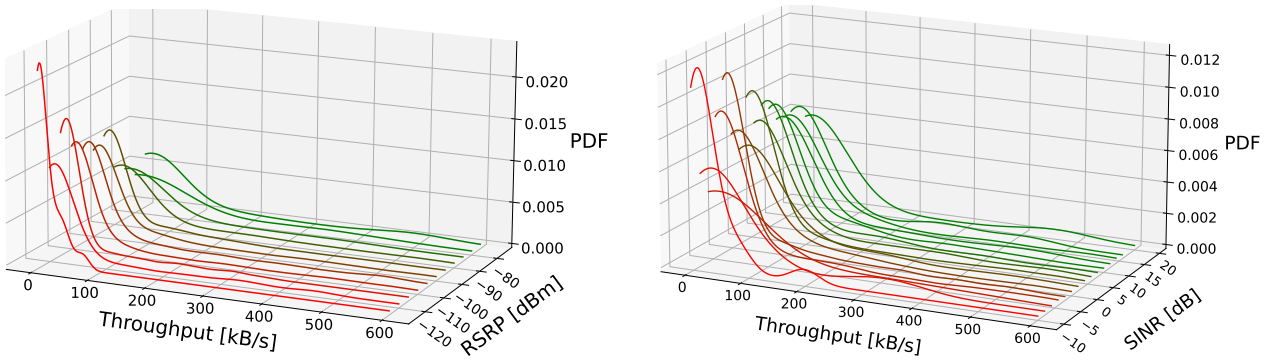
Figure 3.10: Impact on jitter of cell handover among different network technologies

As expected, in the presence of a horizontal handover, all the network technologies analyzed show an increase in their RTT and jitter levels. In the case of vertical handovers, connections starting in 4G LTE networks increased their RTT and jitter to higher values than for horizontal handovers. Contrarily, vertical handovers in connections starting in 2G EDGE networks decreased their RTT and jitter values. However, these behaviors were expected, as a vertical handover starting in a 4G LTE network will always change to a network technology with lower capacity (as during the crowdsourcing campaigns, there was no 5G network deployed in Chile). On the other hand, a vertical handover starting in a 2G EDGE network will change to a network technology with higher capacity. With regard to connections starting in 3G UMTS or 3G HSPAP networks, there is not a significant difference in RTT and jitter values in the presence of horizontal or vertical handovers.

As described above, there is a significant amount of variability that makes it difficult to find a one-to-one correspondence between signal quality and mobile QoS. Therefore, our intuition was that it would be a better approach to study the relationship between signal quality and the probability distribution of mobile QoS. In order to corroborate this intuition, we first calculated the average throughput for all the observed connections (both TCP and UDP) as the ratio between the number of received bytes and the network connection duration (downlink throughput). It is important to note that the computed average throughput may not always be a meaningful performance indicator. If the network flow is not related to a network-intensive application, then the obtained average throughput may not be a good indicator of the channel capacity. Therefore, we filtered the network flows to only those related to the Youtube application, as it was the most used network-intensive application in the dataset. Moreover, and guided by our previous analysis, we considered only the connections established via 4G LTE, in order to get meaningful results.

Figure 3.11 shows the changes in the probability density function (PDF) of the average throughput of Youtube-related connections given by different values of RSRP and SINR. In both cases, there is an evolution of the PDF as the signal quality levels increase. More

precisely, as the RSRP and the SINR increase (better signal quality), the PDF of the average throughput concentrates to higher values. Thus, when the signal quality is better, the random variable (average throughput) is more likely to be close to higher values.



(a) (b)

Figure 3.11: Probability density functions of average throughput according to (a) different RSRP values and (b) different SINR values. The figures only consider Youtube-related connections established via 4G LTE.

Even when Figure 3.11 clearly indicates a tendency, these results are limited by the nature of the collected dataset. In our dataset, each network flow is related to an average throughput and to an average signal quality that summarize the network connection duration (which can last up to several seconds). However, a deeper study regarding these insights is presented in the following chapter.

3.7 Summary

In this chapter, we presented a novel methodology for obtaining periodic passive measurements in user-space without requiring root privileges, called PePa Ping. This methodology allowed us to collect valuable information about network usage, overall empirical QoS of network flows, and a comprehensive set of mobile users' contextual information. Thus, we provide a useful source that can be of great importance to a variety of studies related to network usage behavior and network performance.

After inspecting the collected dataset, we showed that the variability of mobile QoS can be related to different contextual sources besides signal quality. Thus, we were motivated to study the relationship between signal quality and the probability distribution of mobile QoS instead of finding a one-to-one correspondence between signal quality and mobile QoS.

According to our preliminary results, the signal quality does have a clear relationship with the probability distribution of mobile QoS, particularly with the probability distribution of average throughput. Therefore, our following efforts are focused on gaining a deeper understanding of this phenomenon.

Chapter 4

Anticipatory Networking for Mobile QoS

4.1 Introduction

The evolution of mobile networks to newer technologies has brought with it a number of challenges and critical requirements, such as ultra-reliable and low-latency communications, anticipatory resource allocation, bandwidth efficiency, and optimized user QoE. However, one of the major drawbacks in analyzing mobile networks is their highly dynamic behavior, as most of their properties could rapidly change over time. Within this context, there has been an increasing amount of literature on user throughput prediction for anticipatory networking, since accurate throughput predictions are challenging but essential for the efficiency of different networking applications, such as resource scheduling [82, 24, 104] and adaptive video streaming [90, 16, 103, 81, 169]. Certainly, adaptive bitrate algorithms critically rely on throughput estimations to select the proper download bitrate dynamically, and therefore, accurate predictions are essential to optimize user QoE. According to the Ericsson Mobility Report, video streaming constituted around 70% of all global mobile network traffic in 2022 [37]. Thus, throughput prediction has become increasingly important, driven by the continuous development and adoption of bandwidth-demanding applications.

From a theoretical perspective, the channel quality delimits the network performance. Indeed, an upper bound to the throughput (channel capacity) can be defined as a function of the radio link quality (signal-to-interference-plus-noise ratio). Nevertheless, the user throughput is not expected to reach this upper bound in practice, as the empirical measurements are affected by many other dynamic factors, such as user's mobility and speed [71, 134, 115], network congestion [53], and protocol overhead [42].

The intrinsic relation between physical layer metrics and network performance has been previously analyzed using real-world datasets [18, 38, 59, 104, 120, 134, 178, 181, 138]. Interestingly, the relationship between signal quality and user throughput is commonly reported as a weak positive correlation [59, 157, 156, 161, 38]. Also, some research studies have included different signal measurements into supervised machine learning models to improve the

prediction of throughput [136, 137, 148, 178, 135, 168]. However, these prediction models lack further analysis of the practical implication between channel quality metrics and the throughput observed by the user. As mentioned before, the presence of a positive correlation in real data is expected from theory. Although, no comprehensive study has analyzed in-depth the effect of channel quality fluctuations on throughput, putting particular attention on the high variability of user throughput, and how this information can be used to develop an easily explainable throughput prediction model.

In this chapter, we present important contributions to the state-of-the-art in two different directions.

Firstly, we thoroughly analyze the impact of channel quality variations on user throughput using a novel approach. Different from other authors, we study the user throughput as a random variable that depends on the current signal-to-interference-plus-noise ratio (SINR). Thus, we model the distribution of user throughput as an SINR-dependent probabilistic mixture model that properly fits the empirical data. This approach allows a more comprehensive understanding of the empirical effect of the SINR on user throughput, as we could directly apply different concepts of probability distribution theory. Additionally, we present a methodology to extrapolate the probability distribution of user throughput for any SINR value, even for those absent (or poorly represented) in the original dataset.

Secondly, we present two different approaches to predict instantaneous user throughput based on our SINR-dependent probabilistic mixture model. These methodologies use two strategies to estimate the mathematical expectation of the random variable (user throughput). Therefore, our throughput estimators can be directly explainable as they correspond to the expected value of the probability distribution of user throughput. According to our experimental results, these approaches are shown to be valuable for practical throughput prediction applications in different time scales.

4.2 Motivation

From an information theory perspective, it is well known that channel quality delimits the network performance. Indeed, according to Shannon capacity theorem, the channel capacity (C) in bits per second can be defined as a function of the bandwidth (B) in hertz and the signal-to-noise ratio (SNR) of the link:

$$C = B \times \log_2(1 + \mathrm{SNR}), \tag{4.1}$$

where the SNR is the ratio of the received signal power (signal strength) to the power of noise.

For wireless networks, the channel capacity is often modeled as a function of the signal-to-interference-plus-noise ratio (SINR) [24, 124, 121, 140], where the SINR is analogous to the SNR used in (4.1). More precisely, the SINR is the ratio of the received signal power (signal strength) to the sum of the power of the unwanted interference from other signals and the power of noise.

The referenced Shannon's theorem in (4.1) is important as it gives an upper bound for the communication rate. However, this upper limit has been proved to be quite optimistic in realistic scenarios [124]. For practical applications, a more employed term is throughput. The throughput refers to how much data is successfully transferred from source to destination at any given time, and it is measured in bits per second. Commonly, the throughput measures the data rate of a single application session. Additionally, unlike the theoretical channel capacity, the throughput will be affected by protocol overhead and channel contention produced by simultaneous network flows.

User throughput is a critical determinant in the user quality of experience (QoE) [140, 27], and therefore, several authors have addressed the problem of throughput prediction for mobile network users. Accordingly, a number of research studies have included different channel quality metrics in their throughput prediction models, such as the SINR and other closely related physical layer KPIs (RSSI, RSRP, RSRQ, and CQI, among others). Moreover, these metrics are commonly reported to clearly improve throughput prediction [181, 104, 137, 18, 148, 125, 36].

Numerous studies have inspected the empirical correlation between mobile throughput and signal quality metrics, commonly seeking a linear dependence [18, 38, 161]. Indeed, attempting to find a one-to-one correspondence between signal quality and mobile throughput is well-founded in the theory [140, 124] (e.g., Shannon Theorem in (4.1)). Nevertheless, the empirical evidence shows that such a direct correlation does not exist in practice. Several authors have analyzed the correlation between mobile network throughput and different lower-layer metrics [18, 38, 59, 104, 120, 134, 178, 181, 138]. These studies agree on the existence of a positive correlation between channel quality and user throughput; however, it is often reported as a weak correlation [59, 157, 156, 161, 38]. Most of these analyses employ visual representations to examine the relation between these variables, such as scatter plots [38, 178, 138] and box-plots [134, 120, 156, 161]. These visual aids clearly evidence the existence of a relationship between lower-layer information and mobile throughput. However, these representations also reveal the remarkable throughput variability for every channel quality level. Figure 4.1 exemplifies the results obtained by those research studies when inspecting the correlation between throughput and different lower-layer metrics for mobile networks (the color yellow indicates the highest concentration of measurements).

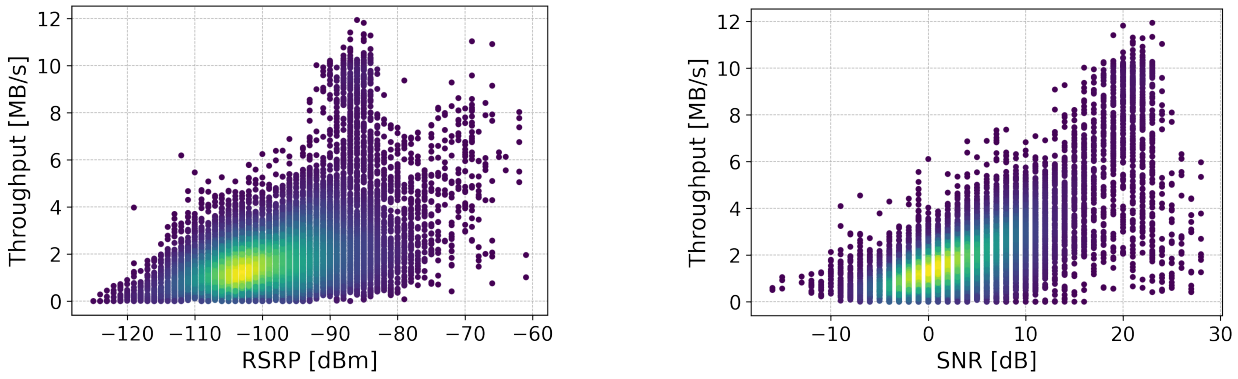


Figure 4.1: Relationship between user throughput and channel quality (RSRP and SNR) in a 4G LTE mobile network. Dataset from Raca et al. [134] (Operator B's network).

Accordingly, the empirical experimentation reveals that none of the lower-layer metrics have a direct functional relationship with user throughput. This difference between theory and practice can be explained by other factors affecting throughput that are not related to channel quality, e.g., protocol overhead, network contention, and those previously discussed in Section 3.6. Although, no comprehensive study has analyzed in-depth the effect of channel quality fluctuations on user throughput, putting particular attention on the high variability of user throughput for each quality level. In this work, we study the user throughput as an SINR-dependent random variable, and therefore, we model its distribution as a probabilistic mixture model based on the current SINR value. Furthermore, we use this information to develop an explainable throughput prediction model for mobile users based on different concepts of probability distribution theory.

4.3 Background

Much of the current literature on throughput prediction pays particular attention to aiding adaptive video streaming. Adaptive bitrate algorithms critically rely on throughput predictions, and thus, accurate throughput prediction is essential to improve user experience [137, 16, 176, 81, 82]. However, throughput prediction in wireless networks is particularly challenging given the rapidly varying network conditions. Consequently, many authors have addressed the problem of throughput prediction for mobile users by proposing different methodologies.

Xu et al. [176] proposed PROTEUS, a regression tree-based methodology for throughput prediction in 3G mobile networks. PROTEUS collects network measurements such as loss rate, one-way delay, and throughput to predict future network performance in time windows of 500ms. According to the authors, PROTEUS can assist in improving the perceived quality in video conferencing scenarios.

Liu et al. [86] presented a systematic study on the comparative performance of different throughput prediction techniques for 3G mobile networks. The authors employed seven prediction approaches and analyzed the effect of different variables, such as prediction horizon and geographic location. Additionally, the authors applied an entropy-based approach to obtain an estimated lower bound on throughput prediction errors.

Yue et al. [178] introduced LinkForecast, a random forest-based framework to predict throughput in 4G mobile networks. LinkForecast relies on historical throughput measurements and different lower-layer metrics (RSRP, RSRQ, CQI, BLER, and handover events). The authors collected the channel quality metrics directly from mobile devices (low fidelity) and from specialized tools (high fidelity). According to their results, both approaches lead to similar prediction accuracy. Therefore, LinkForecast is proposed as a reliable methodology to be implemented over regular mobile devices.

Samba et al. [148] proposed a history-less throughput prediction for 4G LTE mobile networks. The authors were interested in predicting throughput before a connection is established, when previous throughput values are unavailable. The prediction methodology employed the random forest algorithm with a comprehensive set of contextual information,

including channel quality metrics (RSRP and RSRQ), user mobility (indoor/outdoor, distance to cell, and speed), and in-cell measurements (cell throughput, number of users on the cell, among others).

Oussakel et al. [125] deployed a 4G LTE testbed to predict uplink throughput based on two machine learning techniques (random forest and support vector machines). The proposed algorithms utilize as input a set of radio metrics taken by the eNodeB base station (RSSI, SINR, received power, among others). Their results show that uplink throughput predictions are less accurate than downlink throughput predictions.

Zhohov et al. [181] proposed a throughput prediction method for 4G mobile networks based on gradient-boosting decision trees. The proposed method utilized information from the user-side (RSRP, RSRQ) and from the network-side (cell logs). Additionally, the authors highlighted the importance of the explainability of throughput prediction models, as it is crucial for the adoption of the proposed methods.

Eyceyurt et al. [38] analyzed different methodologies for throughput prediction in 4G mobile networks based on physical layer metrics (RSRP, RSRQ, and SNR). The authors used five machine learning algorithms to predict uplink throughput in different environments (urban, suburban, and rural areas). According to their results, the decision tree and k-nearest neighbor algorithms achieved the highest prediction accuracy.

Minovski et al. [104] proposed a machine learning-based throughput prediction method for 4G LTE and 5G mobile networks. The authors aimed to develop a non-intrusive model without using previous throughput measurements, but only passive metrics (RSSI, RSRP, SINR, frequency band, and cell load, among others). Thus, the selected features are utilized to directly predict user throughput using different supervised machine learning techniques, such as decision tree-based algorithms and deep neural networks.

Boban et al. [18] studied the problem of throughput prediction for vehicular use cases that require stable and high uplink throughput. The authors employed different machine learning approaches, such as linear regression, random forest, and deep neural networks. All models were trained using channel quality metrics (RSRP and SINR), geographic location, and user speed. As reported by their results, the best indicator of uplink throughput is SINR.

Raca et al. [137] proposed a throughput prediction method for 4G LTE mobile networks based on the random forest algorithm. The prediction model relies on a novel summarization of historical throughput and radio channel metrics (RSRQ, RSRP, CQI, and SINR), outperforming other classical throughput estimation techniques such as the exponentially weighted moving average (EWMA). Additionally, the authors discussed the use of more sophisticated learning models such as LSTM neural networks, concluding that they can be helpful to enhance the learning of rare network scenarios [135].

Narayanan et al. [115] presented Lumos5G, a machine learning framework to predict user throughput in 5G mobile networks. The authors conducted an exhaustive feasibility study of predicting mmWave 5G throughput based on information from the user-side (channel quality metrics, movement pattern, speed, geographic location) and from the network-side (5G panel configuration).

Elsherbiny et al. [36] studied the performance of different techniques for throughput prediction in a 4G LTE network along a public transportation route. The authors explored the performance of classical machine learning models (support vector regression, k-nearest neighbor, random forest, among others) and time series forecasting models (ARIMA and LSTM neural networks). The models base their predictions on a set of contextual information obtained in user-space (RSRP, RSSI, SNR, RSRQ, timestamp, longitude, latitude, and historical throughput values). Among all the throughput prediction techniques, the random forest algorithm achieved the highest prediction accuracy.

Wei et al. [168] proposed TRUST, an LSTM-based throughput prediction method for 4G LTE mobile networks. During its first stage, TRUST collects information from the user device (RSSI, TCP throughput, Cell ID, time, and user location, among others) to identify a movement pattern (static, walk, bus, and train scenarios). Then, TRUST employs an LSTM model specifically trained for the movement pattern identified to predict future throughput. Each LSTM model receives as input the same information as in the movement pattern identification stage. According to the authors, TRUST outperforms other more straightforward throughput prediction methods, highlighting the importance of user movement pattern identification before prediction.

Na et al. [114] proposed an attention-based LSTM model for throughput prediction in 4G LTE mobile networks. This method predicts user throughput based solely on the previous throughput measurements. According to the authors, the LSTM model with the attention method shows higher throughput prediction accuracy than the LSTM model without the attention method.

Biernacki et al. [16] presented an LSTM-based throughput prediction methodology for 4G and 5G mobile networks. This method utilizes historical throughput measurements to predict throughput for the next four seconds. According to their results, the proposed prediction approach improves video quality for different DASH players.

Azmin et al. [11] proposed a Transformer-based mechanism to predict user throughput in 5G mobile networks. The authors employed the Informer model with an initial machine learning-based feature selection step. The model base its predictions on a comprehensive set of contextual information such as GPS coordinates, speed, timestamp, cell id, and channel quality metrics (SNR, RSRP, RSRQ, RSSI, and CQI, among others).

Lekharu et al. [81] presented an LSTM-based throughput prediction model for 4G LTE mobile networks. According to the authors, the adoption of the proposed method for DASH video improves user QoE, as the LSTM model outperforms other throughput prediction strategies such as random forest and linear regression.

4.3.1 Limitations of previous works

More recent attention has focused on the proposal of deep learning-based approaches for throughput prediction, mainly based on LSTM neural networks [168, 114, 16, 36, 11, 81, 169, 82]. Nevertheless, there is no conclusive evidence that these new methods outperform the more straightforward methods previously proposed [81, 36, 135, 104, 18]. Indeed, the

results obtained by Liu et al. [86] had already revealed that more complex algorithms are not necessarily better for predicting user throughput in mobile networks. Moreover, most of the proposed deep learning approaches lack a discussion on the explainability of the models, being difficult to analyze the effect of the selected features on the prediction of user throughput, which is essential for the adoption of the methodologies [181]. In this work, we propose two different approaches to predict user throughput based on the empirical effect of SINR. These methodologies are based on the probability distribution theory and can be easily interpreted as they directly estimate the mathematical expectation of user throughput.

Some research works have claimed the importance of history-less throughput prediction methodologies, i.e., to predict throughput when previous throughput values are unavailable [148]. These prediction strategies can be essential not only when a connection has not been established yet, but also after every vertical handover, where the user device changes its connectivity from one mobile network technology to another. In the latter case, previous throughput values may not be appropriate to predict user throughput in the current network technology. Thus, we present in Section 4.8.1 an initial version of our SINR-dependent probabilistic model for throughput prediction that only relies on the current SINR value without any past throughput information.

Previous studies have explored the benefits of including contextual information in throughput prediction methods. Nevertheless, some of these features may not be available in practical in-user applications, such as precise GPS coordinates [18, 115, 36, 11], information about user mobility patterns [168, 115, 148], user speed [18, 115, 148, 11], and network-side measurements (cell load, number of users on the cell, cell logs, among others) [114, 104, 115, 148, 125]. Our proposed approach relies on a unique contextual metric to predict user throughput: the SINR. The SINR is usually directly available from the mobile operating system, and thus, it has been commonly included in network datasets for throughput prediction [11, 104, 115, 125, 134, 139]. Consequently, our prediction methodology can be deployed and used by applications running on regular commercial mobile devices.

Moreover, as mentioned by Liu et al. [86], most state-of-the-art throughput prediction methodologies are not directly comparable as their underlying assumptions differ. These differences are mainly determined by the network dataset each study has access to, and therefore, the proposed prediction strategies are not trivially transferable to another mobile context. In this work, we test our proposed throughput prediction methodologies in four open mobile network datasets, covering a wide variety of measurement contexts. Thus, our approaches are shown to be functional and applicable to different mobile scenarios, including different network technologies, countries, measurement methodologies, and sizes of the network area covered.

4.4 Dataset Overview

In order to perform a representative study of realistic mobile network scenarios, we aimed to employ empirical data to: (*i*) understand the effect of SINR on user throughput as perceived by applications running on the user equipment, and (*ii*) predict throughput using an SINR-dependent approach. Therefore, we sought public datasets containing fine-grained mobile

network measurements about SINR and user throughput. In the following, we describe each of the four selected datasets, which all were collected by running experiments on operational mobile networks based on different technologies all over the world. Additionally, these studies used different methodologies to measure downlink throughput, even measuring throughput at different network layers (data-link, transport, and application). All these differences among the selected datasets help support the robustness of our later analysis.

4.4.1 IMC'20-Lumos5G dataset [115]

- **Mobile network description:** All measurements were taken from an operational mmWave 5G network in Minneapolis, United States.
- **Measurement methodology:** Network testing from four smartphones running a custom application for 5G performance monitoring and throughput measurement (based on iPerf). Network measurements were taken in different mobility modes (walking, driving, and stationary), covering more than 450 km.
- **Throughput measurements:** Downlink throughput is measured at the transport layer (Transmission Control Protocol). Network traffic is generated by using the iPerf network testing tool to measure network throughput.
- **Total number of samples:** ∼560,000
- **Time-resolution:** One second

4.4.2 DataInBrief'22-4G dataset [62]

- **Mobile network description:** All measurements were taken from seven different eNodeBs in an operational 4G LTE network in Lagos State, Nigeria.
- **Measurement methodology:** Drive testing using a 4G LTE modem connected to a computer. Network measurements were taken at a near-constant speed of 30 km/h during a period of almost 12 hours.
- **Throughput measurements:** Downlink throughput is measured at the data-link layer level (Packet Data Convergence Protocol). In the 4G LTE protocol stack, the PDCP layer is located above the IP layer. Network traffic is generated by simultaneously downloading several 20 GB files over File Transfer Protocol (FTP).
- **Total number of samples:** ∼40,000
- **Time-resolution:** One second

4.4.3 GLOBECOM'20-4G dataset [139]

- **Mobile network description:** All measurements were taken from a single eNodeB in an operational 4G LTE network in Vienna, Austria.

- **Measurement methodology:** Static indoor testing from a single smartphone using the Nemo Handy application from Keysight Technologies, Inc. Network measurements were taken during a period of 90 hours.
- **Throughput measurements:** Downlink throughput is measured at different layers: physical, data link, and application. To embrace diversity in our empirical analysis, we considered in this case the application layer throughput (HTTP), which does not include TCP/IP headers. Network traffic is generated by downloading a 40 GB file over HTTP.
- **Total number of samples:** ∼600,000
- **Time-resolution:** 500 milliseconds

4.4.4 MMSys'18-4G dataset [134]

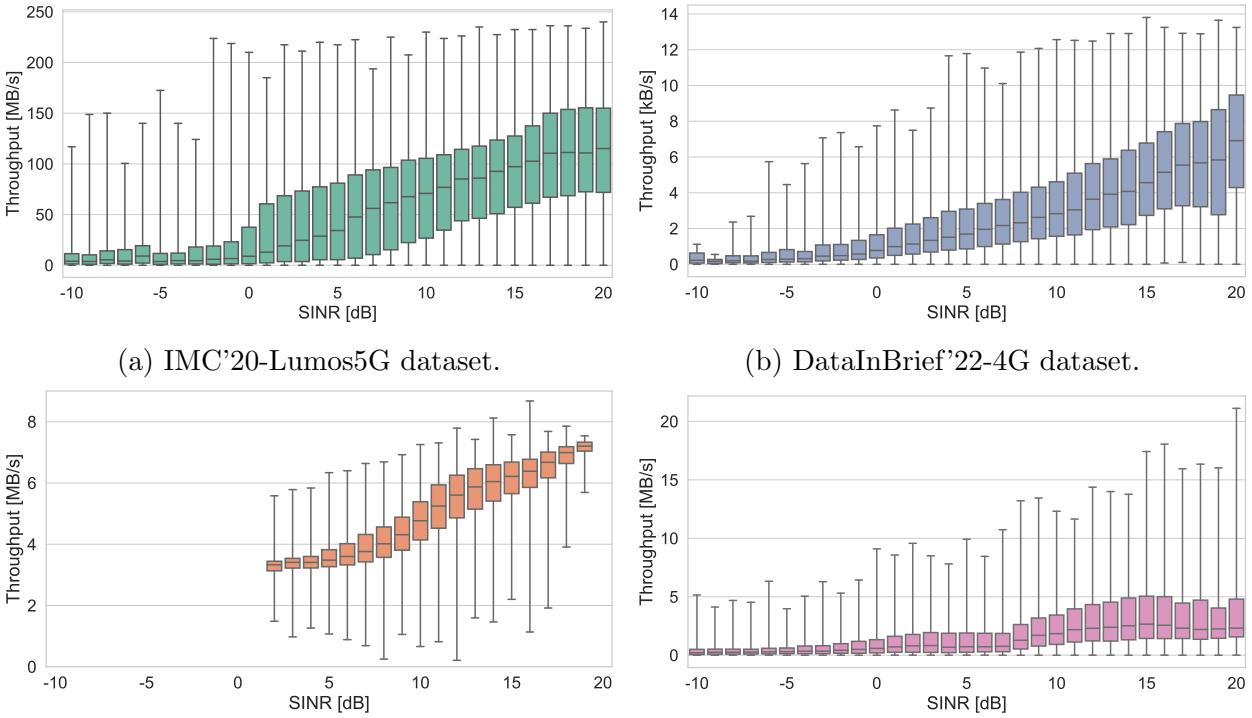
- **Mobile network description:** Network measurements were taken from two operational 4G LTE networks in Irland (Operator A and Operator B).
- **Measurement methodology:** Network testing from three different smartphones running the G-NetTrack Pro mobile network monitoring tool. Network measurements were taken in different mobility modes (static, pedestrian, bus, car, and train).
- **Throughput measurements:** Downlink throughput is measured at the application layer by the G-NetTrack Pro mobile application. Network traffic is generated by downloading large files (connection-oriented, using TCP).
- **Total number of samples:** ∼140,000, divided into 101 traces (Operator A)
  ∼30,000, divided into 34 traces (Operator B).
- **Time-resolution:** One second

In Sections 4.5, 4.6, and 4.7, we use the four previously described datasets to thoroughly study the effect of the SINR on user throughput, and to elaborate a probabilistic mixture model that estimates the probability density function of throughput as a function of the current SINR value. In Section 4.9, we test our proposed SINR-dependent probabilistic mixture model to predict throughput for mobile networks using a portion of the MMSys'18-4G dataset. We utilize the measurements corresponding to Operator B's mobile network, which were not used in the previous sections.

4.5 Throughput Dependence on SINR

As mentioned in Section 4.2, we attempted to revisit the problem of analyzing the empirical dependence of mobile user throughput on the SINR. Thus, we employed the open datasets described in Section 4.4 to provide a more reliable analysis. Figure 4.2 shows the empirical

impact of the SINR on the observable throughput in the four real scenarios previously selected. As expected, all cases show a clear positive correlation between SINR and median throughput. This positive correlation has been previously used to fit regression models for the throughput based on the SINR [18, 38]. However, attempting to model throughput as a function of the SINR is not sound, as there is no realistic one-to-one correspondence between SINR and throughput values. As shown in Figure 4.2, there is a high variability of throughput for each SINR value, and this information should be taken into account.



(a) IMC'20-Lumos5G dataset.

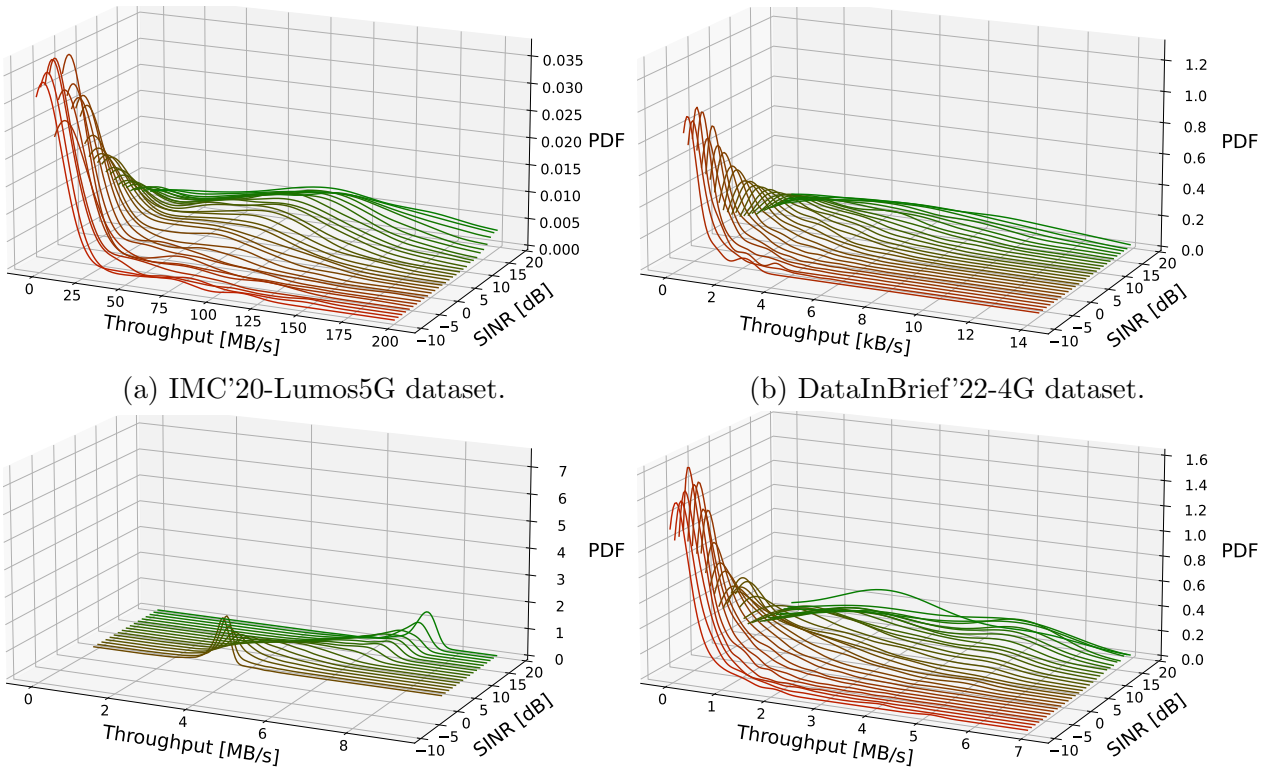
(b) DataInBrief'22-4G dataset.

(c) GLOBECOM'20-4G dataset.

(d) MMSys'18-4G dataset.

Figure 4.2: Box-plots showing the changes in the distribution of user throughput as the SINR varies.

We aimed to better understand the impact of SINR on the observable throughput measurements without limiting the empirical data. Accordingly, our intuition was that the effect of channel quality on throughput could be better understood if we analyzed the impact of signal quality on the probability distribution of throughput. Figure 4.3 shows the changes in the probability density function of user throughput given by different SINR values. These charts clearly exhibit an evolution of the probability density function (PDF) of throughput as the SINR value increases. Regardless of their differences, this probability distribution progression is evident for the four selected datasets. In the following, our efforts focus on further understanding these probability distribution progressions.



(a) IMC'20-Lumos5G dataset.

(b) DataInBrief'22-4G dataset.

(c) GLOBECOM'20-4G dataset.

(d) MMSys'18-4G dataset.

Figure 4.3: Empirical probability density functions of user throughput according to different SINR values

4.6 Gamma-Gaussian Mixture Model

Based on our previous dataset inspection, we aim to describe the probability density of user throughput as a function of SINR. Accordingly, Figure 4.3 evidenced the following relationship for the four datasets: as the SINR increases, the PDF of user throughput progressively changes. The four empirical probability density progressions shown in Figure 4.3 suggested that each probability density function is actually comprised of two components. The first component is accentuated at lower SINR values, where the population of throughput measurements tends to values closer to 0. Intuitively, this first component is related to the lower bound of possible throughput values. By definition, user throughput can never be negative, and therefore, there is a higher probability density concentration near 0 as the SINR decreases. The second component becomes more noticeable at higher SINR values (when the first component starts to fade out). Similarly, this component is related to the upper bound of possible throughput values. However, this limit is not regulated only because of the theoretical and physical limits imposed by the mobile network technologies, but also because of any network bandwidth management scheme used by the network operators. Therefore, this limit is more diffuse, and the probability density does not evidently concentrate on a single value as the SINR increases.

Based on the previously described two-component behavior of the PDF of user throughput, we decided to model the probability distributions using a Gamma-Gaussian Mixture

Model. A Gamma-Gaussian Mixture Model (GGMM) is a 5-parameter probabilistic model whose corresponding probability density function is given by the mixture of a Gamma distribution and a Gaussian distribution, i.e.,

$$f_{\text{GGMM}}(x;\, \alpha, \beta, \mu, \sigma, \lambda) := \lambda f_1(x;\, \alpha, \beta) + (1 - \lambda) f_2(x;\, \mu, \sigma), \tag{4.2}$$

where $f_1(\cdot\,;\, \alpha, \beta)$ is the PDF of the Gamma distribution with parameters $\alpha, \beta > 0$, defined as

$$f_1(x;\, \alpha, \beta) := \frac{x^{\alpha-1} e^{-\beta x} \beta^{\alpha}}{\Gamma(\alpha)}, \tag{4.3}$$

where $\Gamma(\cdot)$ is the Gamma function; and $f_2(\cdot\,;\, \mu, \sigma)$ is the PDF of the Normal distribution with parameters μ and $\sigma > 0$, defined as

$$f_2(x;\, \mu, \sigma) := \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}; \tag{4.4}$$

and $\lambda \in [0, 1]$ adjusts the weight of both mixture components.

Before employing the Gamma-Gaussian Mixture models, we took the four datasets described in Section 4.4 and split them into training and testing sets, with a 75:25 split ratio. In this section, we only utilize the training sets to study the goodness of the Gamma-Gaussian Mixture models to describe user throughput. The testing sets are used later in Section 4.7 for a secondary validation.

For each training set, we group the user throughput measurements by their SINR value. Then, we computed the empirical PDF of user throughput for each group, as previously shown in Figure 4.3. Finally, we fit a GGMM to each of these empirical PDFs. Figures 4.4-4.7 exemplify the fit of the GGMM to different levels of SINR in the four datasets (low, medium, and high SINR values). The figures clearly illustrate how the two components of the GGMM adapt to properly capture the probability density of user throughput as the SINR changes. Furthermore, Figure 4.8 summarizes the coefficient of determination (R^2) obtained in the four datasets at fitting the GGMM to the empirical throughput distribution corresponding to every SINR level. These values confirm the goodness of this fitting model, as the median R^2 score is higher than 0.95 for all the datasets.

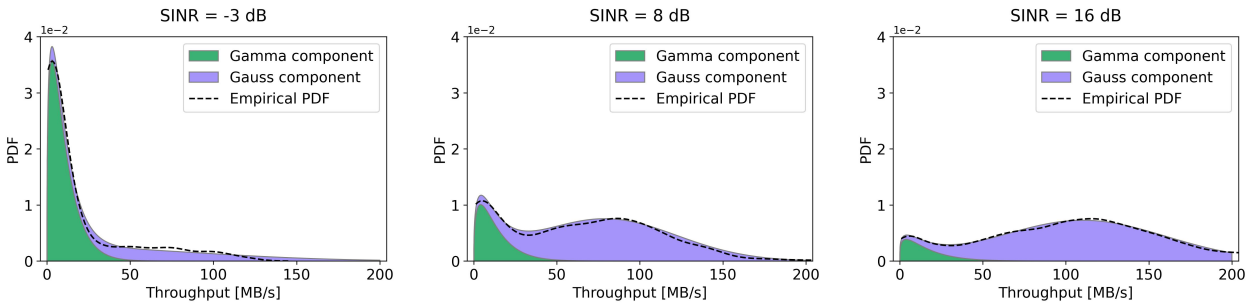


Figure 4.4: IMC'20-Lumos5G dataset: fitted GGMMs to the probability density function of user throughput for selected SINR values (-3 dB, 8 dB, and 16 dB).

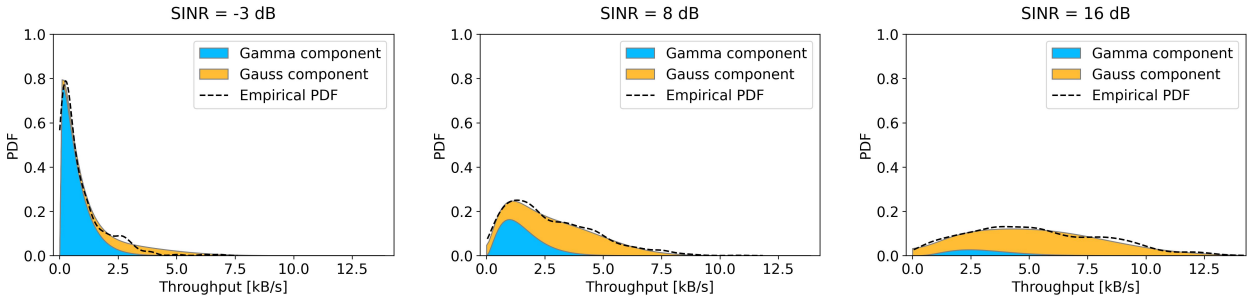


Figure 4.5: DataInBrief'22-4G dataset: fitted GGMMs to the probability density function of user throughput for selected SINR values (-3 dB, 8 dB, and 16 dB).

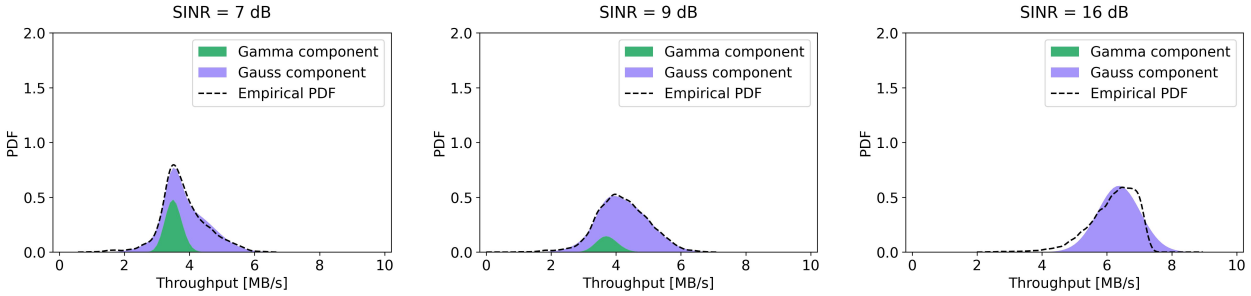


Figure 4.6: GLOBECOM'20-4G dataset: fitted GGMMs to the probability density function of user throughput for selected SINR values (7 dB, 9 dB, and 16 dB).

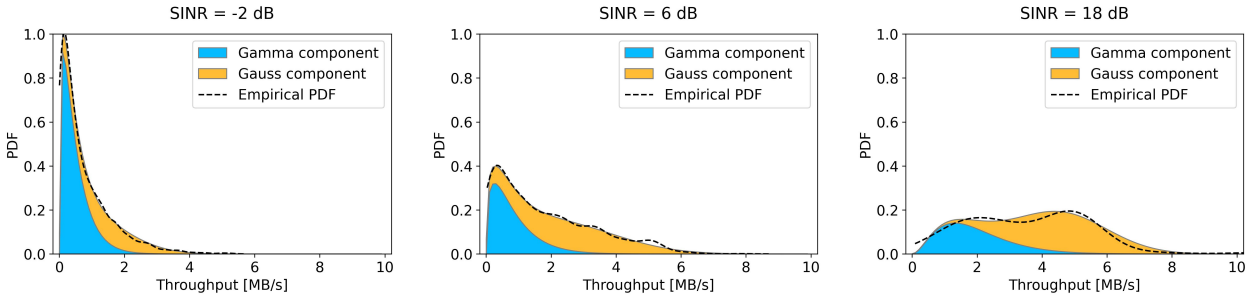


Figure 4.7: MMSys'18-4G dataset: fitted GGMMs to the probability density function of user throughput for selected SINR values (-2 dB, 6 dB, and 18 dB).

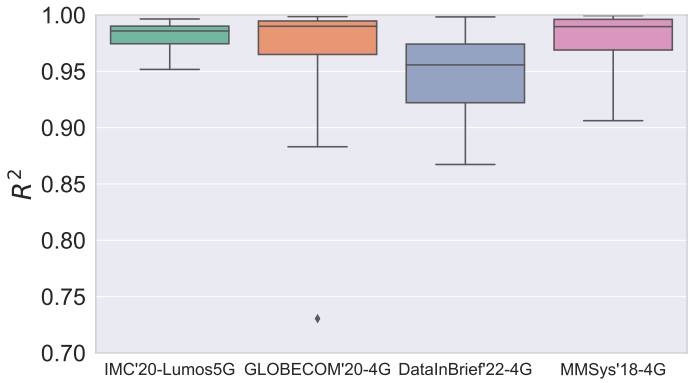


Figure 4.8: Box-plots showing the R^2 values obtained by fitting the GGMM to the empirical PDF of user throughput according to every SINR value.

4.7 Estimating the Probability Density Function of User Throughput

In the previous sections, we have studied (i) the dependence of user throughput on the SINR value and (ii) the goodness of fitting an SINR-dependent Gamma-Gaussian Mixture Model to estimate the probability density of user throughput. However, a major challenge would be to estimate the PDF of user throughput for any SINR value, even for those absent (or poorly represented) in the original datasets. Therefore, in this section, we look into an extrapolation methodology for our previous results, in order to be able to provide a GGMM that characterizes user throughput for any particular SINR level.

As the GGMM is a 5-parameter model, we can study the evolution of the PDFs of user throughput by analyzing the evolution of these five parameters. We aim to apply a regression model to these parameters to estimate their relationship with the SINR (independent variable). Thus, we could predict the probability density function of user throughput for any SINR value by extrapolating these five parameters. This PDF estimation would not only be beneficial to describe user throughput for missing SINR values in the datasets, but also to provide a more reliable PDF of user throughput for SINR values already present in the datasets.

Before applying a regression model to the five parameters of the GGMM, we filter the fitted models obtained in Section 4.6 to only those having an R^2 value of more than 0.9. Additionally, for the gamma-component parameters (α and β), we only took into account the fitted GGMMs with parameter $\lambda > 0.1$. Analogous, for the gaussian-component parameters (μ and σ), we only took into account the fitted GGMMs where $(1 - \lambda) > 0.1$. We selected these filter thresholds to avoid using improper values in the regression models.

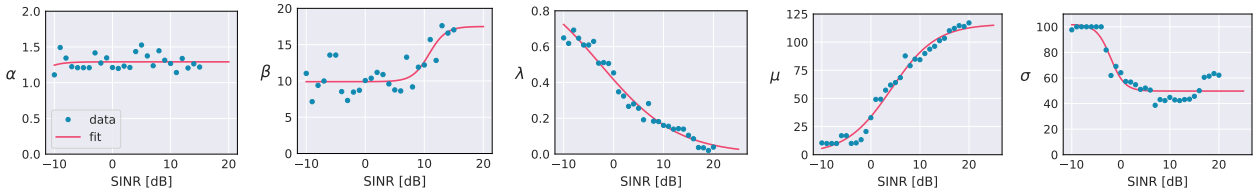
As described in Section 4.6, the empirical PDFs of user throughput suggested an asymptotic behavior in function of the SINR, i.e., the probability density function of user throughput stabilizes for both low and high SINR values. Therefore, we decided to model the relationship between the five GGMM parameters and the SINR by applying a generalized logistic regres-

sion, where the data is fitted by a function constrained by a pair of horizontal asymptotes as $x \to \pm\infty$. Accordingly, for each parameter, the regression model arrives at an equation for the generalized logistic function $y(x)$ that best fits the data, defined as

$$y(x) = y_L + \frac{y_R - y_L}{1 + e^{-k(x-x_0)}}, \tag{4.5}$$

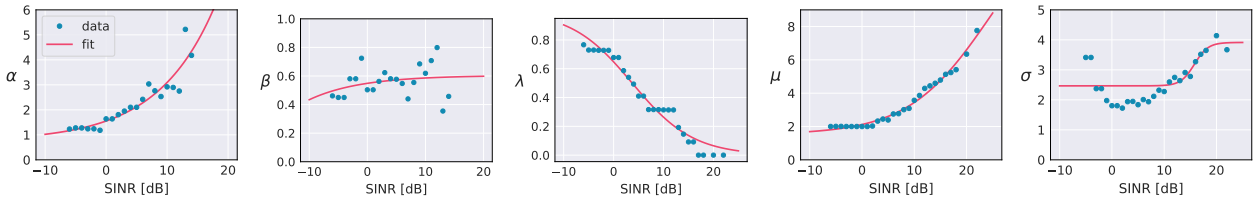
where y_L is the left asymptote, y_R is the right asymptote, k is the logistic growth rate, and x_0 is the sigmoid's midpoint.

Figures 4.9-4.12 exhibit the fitted regression models for the 5 GGMM parameters in each of the four datasets. These figures evidence the suitable selection of the regression model, as it properly captures the relationship between the SINR and the different parameters.



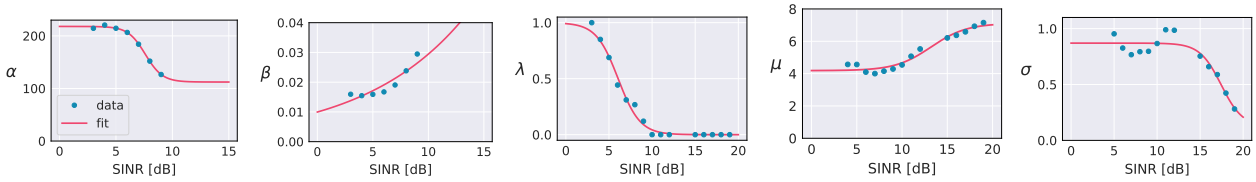
(a) α parameter. (b) β parameter. (c) λ parameter. (d) μ parameter. (e) σ parameter.

Figure 4.9: IMC'20-Lumos5G dataset: Generalized logistic regressions to describe the relationship between the five GGMM parameters and the SINR.



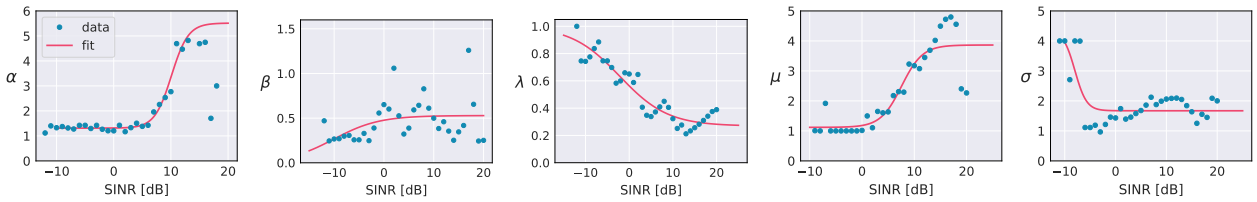
(a) α parameter. (b) β parameter. (c) λ parameter. (d) μ parameter. (e) σ parameter.

Figure 4.10: DataInBrief'22-4G dataset: Generalized logistic regressions to describe the relationship between the five GGMM parameters and the SINR.



(a) α parameter. (b) β parameter. (c) λ parameter. (d) μ parameter. (e) σ parameter.

Figure 4.11: GLOBECOM'20-4G dataset: Generalized logistic regressions to describe the relationship between the five GGMM parameters and the SINR.



(a) α parameter. (b) β parameter. (c) λ parameter. (d) μ parameter. (e) σ parameter.

Figure 4.12: MMSys'18-4G dataset: Generalized logistic regressions to describe the relationship between the five GGMM parameters and the SINR.

According to these results, for any SINR value s, we can predict the probability distribution of user throughput by evaluating the regression models at s. More intuitively, we can consider the function F that takes an SINR value s as input and returns a 5-tuple containing the five parameters of the GGMM associated with the SINR value, i.e.,

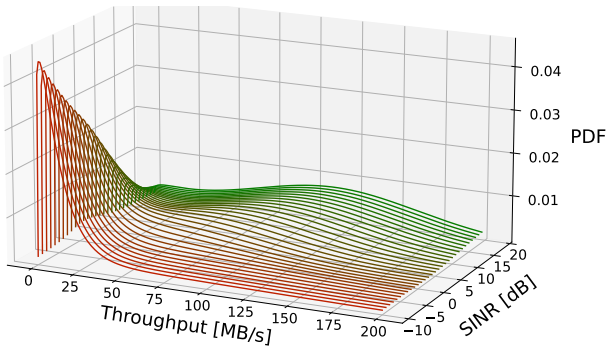
$$F(s) = \Big(\alpha(s),\ \beta(s),\ \mu(s),\ \sigma(s),\ \lambda(s)\Big). \tag{4.6}$$

It is worth mentioning that, given the observable dependence between each of the five parameters and the SINR value (Figures 4.9-4.12), we modeled $F(s)$ as a set of five single-output regressions. Nevertheless, the five parameters of the GGMM could also be estimated by modeling $F(s)$ as a multiple-output regression, where the dependency among the five parameters is implicitly assumed in the model.

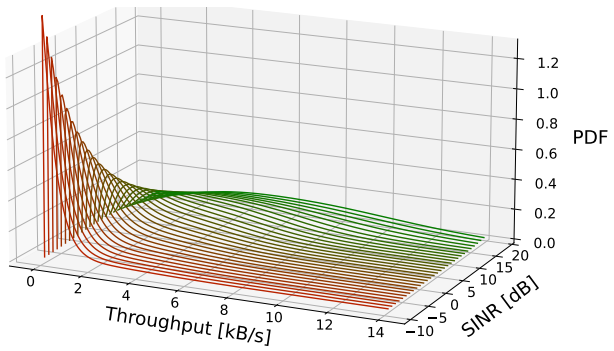
Then, we can employ the function $F(s)$ to estimate the probability density function of user throughput in a wide range of SINR values, even if these values were not present in the original data. For each of the four datasets, Figure 4.13 shows the estimated PDFs of user throughput for SINR values between -10 dB and 20 dB.

These PDFs progressions smoothly capture and extend the empirical PDFs progressions previously observed in Figure 4.3. This extrapolation is particularly significant for the GLOBECOM'20-4G dataset, since it originally contains a small set of SINR values.

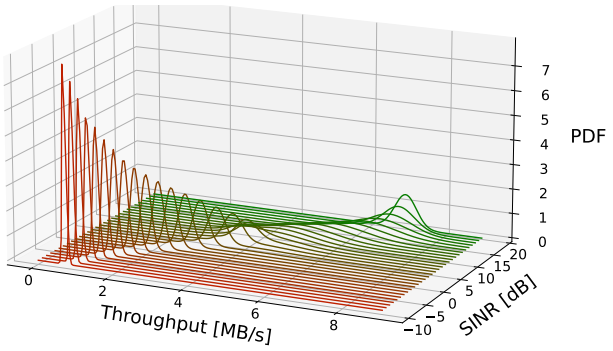
In order to validate the estimated PDFs of user throughput shown in Figure 4.13, we used the testing sets, which had not been utilized so far. Indeed, we calculated the empirical PDFs of user throughput using the testing sets, and therefore, we analyzed how well the GGMM predicts those empirical PDFs. Figure 4.14 summarizes the coefficient of determination (R^2) obtained in the four datasets in predicting the empirical PDFs (testing sets). These values validate the usage of our methodology to estimate the probability distribution of user throughput, as the median R^2 score is higher than 0.85 in each of the four datasets. The obtained R^2 values reveal that the proposed SINR-dependent probabilistic mixture model can be used to properly estimate the PDF of user throughput in a wide variety of mobile network scenarios.



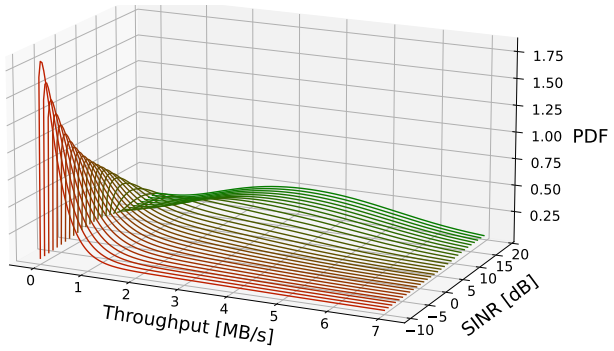
(a) IMC'20-Lumos5G dataset.



(b) DataInBrief'22-4G dataset.



(c) GLOBECOM'20-4G dataset.



(d) MMSys'18-4G dataset.

Figure 4.13: Estimated probability density functions of user throughput according to different SINR values.

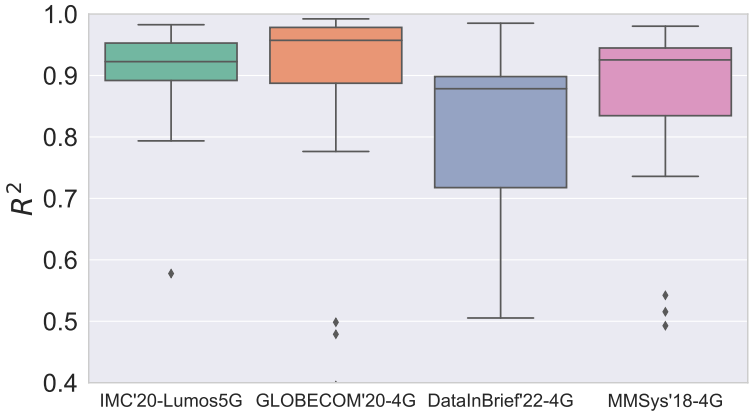


Figure 4.14: Box-plots showing the R^2 values obtained by using the GGMM to predict the empirical probability density functions (testing sets) according to every SINR value.

4.8 User Throughput Prediction

As user throughput is a time-dependent variable, the problem of throughput prediction can be naturally studied as a time series forecasting problem. In the following, we will consider

the user throughput time series $\{X_n\}_{n=1}^{\infty} \subset \mathbb{R}_+$ given by

$$X_n = \text{User throughput at time } n. \tag{4.7}$$

Next, we present two different approaches for predicting the user throughput time series relying on our previous analysis. The first approach is a passive method, as it predicts user throughput based solely on the actual SINR value. The second approach extends the first methodology by incorporating information on the last SINR and last throughput values. Both approaches rely on our previous examinations, and therefore, they assume a prior definition of the function F in (4.6) based on empirical measurements.

4.8.1 Prediction based solely on SINR

Analogous to the user throughput time series, consider the time series $\{Y_n\}_{n=1}^{\infty} \subset \mathbb{Z}$ given by

$$Y_n = \text{SINR value at time } n. \tag{4.8}$$

As discussed in Section 4.7, we can model the probability distribution of X_n as a Gamma-Gaussian Mixture Model with parameters $F(Y_n)$, i.e., for each time $n \in \mathbb{N}$, it holds that

$$X_n \sim \text{GGMM}\big(F(Y_n)\big), \tag{4.9}$$

where $F(Y_n) = \big(\alpha(Y_n),\ \beta(Y_n),\ \mu(Y_n),\ \sigma(Y_n),\ \lambda(Y_n)\big)$.

Thus, for each time n, we have a well-defined probability distribution of user throughput X_n, and therefore, we can easily calculate some important properties such as the mathematical expectation, the variance, and the entropy of user throughput.

However, for practical applications such as adaptive bitrate algorithms, it is crucial to estimate user throughput by providing a unique value. Therefore, in this first approach, we selected the mathematical expectation $\mathbb{E}(X_n)$ as our throughput prediction. As the random variable X_n is modeled by a GGMM with parameters $F(Y_n)$, we have that

$$\mathbb{E}(X_n) = \mathbb{E}\Big(\text{GGMM}\big(F(Y_n)\big)\Big). \tag{4.10}$$

Then, the expected value of the GGMM can be calculated as the weighted sum of the expected values of its gamma and gaussian components:

$$\mathbb{E}\Big(\text{GGMM}\big(F(Y_n)\big)\Big) = \lambda(Y_n)\frac{\alpha(Y_n)}{\beta(Y_n)} + (1-\lambda(Y_n))\mu(Y_n). \tag{4.11}$$

4.8.2 Prediction based on SINR and previous throughput

As discussed in Section 4.3, many throughput prediction approaches utilize part of the previous throughput values $\{X_1, \ldots, X_{n-1}\}$ to predict the next value X_n. Thus, we extended

our first approach by employing information about the previous throughput value. At time n, we already know the outcome value $\bar{x}$ of the random variable X_{n-1}, i.e., we know the user throughput observed in the last time. Therefore, we propose a new throughput estimator based on the conditional expectation of X_n given the event $X_{n-1} = \bar{x}$, defined as

$$\mathbb{E}\big(X_n | X_{n-1} = \bar{x}\big) = \int_0^\infty f_{X_n|X_{n-1}}(x|\bar{x}) x dx, \tag{4.12}$$

where $f_{X_n|X_{n-1}}$ is the conditional probability distribution, which satisfies that

$$f_{X_n|X_{n-1}}(x|\bar{x}) = \begin{cases} \dfrac{f_{X_n,X_{n-1}}(x,\bar{x})}{f_{X_{n-1}}(\bar{x})} & \text{if } f_{X_{n-1}}(\bar{x}) > 0, \\ \\ 0 & \text{if } f_{X_{n-1}}(\bar{x}) = 0; \end{cases} \tag{4.13}$$

where $f_{X_n,X_{n-1}}(x,\bar{x})$ is the joint probability distribution. The probability density function $f_{X_{n-1}}(\bar{x})$ is modeled by the GGMM with parameters $F(Y_{n-1})$, and it can be calculated as the previous SINR value Y_{n-1} is known. To estimate the joint distribution $f_{X_n,X_{n-1}}(x,\bar{x})$, we use the empirical joint distribution, which we denote by $\overline{f}_{X_n,X_{n-1}}$. This empirical joint distribution must be previously calculated based on prior empirical measurements. Then, we approximate the conditional expectation of X_n by

$$\mathbb{E}\big(X_n | X_{n-1} = \bar{x}\big) \approx \int_0^\infty \frac{\overline{f}_{X_n,X_{n-1}}(x,\bar{x})}{\text{GGMM}\left(F(Y_{n-1})\right)} x dx. \tag{4.14}$$

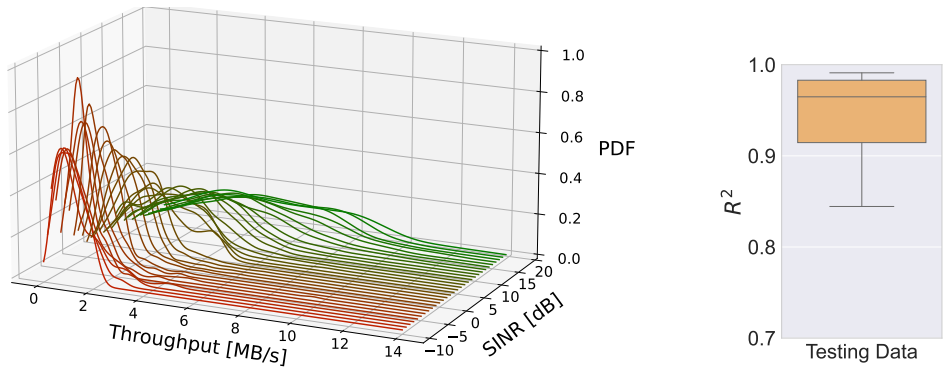
Notice that $\overline{f}_{X_n,X_{n-1}}(x,\bar{x})$ tends to zero when x tends to infinity. Thus, the integral in (4.14) can be numerically computed using a small enough δ value, and a large enough N value such that $N\delta$ is the largest possible throughput value that we consider in the approximation. Finally, in function of N and δ, we get the estimation of the conditional expectation by calculating

$$\mathbb{E}\big(X_n | X_{n-1} = \bar{x}\big) \approx \sum_{i=0}^{N} \frac{\overline{f}_{X_n,X_{n-1}}(i\delta,\bar{x})}{\text{GGMM}\left(F(Y_{n-1})\right)} \cdot i\delta^2. \tag{4.15}$$

4.9 Experimental Results

In this section, we apply all our previous research to provide further validation of our analysis. Therefore, we demonstrate the whole process of using an SINR-dependent probabilistic mixture model to predict throughput for mobile network users. We based these experimental results on a set of data that was not used in the previous sections. This dataset comprises 34 sampling traces corresponding to Operator B in the dataset provided by Raca et al. [134]. As we are interested in analyzing user throughput time series, we divided the dataset into training and testing sets without segregating the sampling traces. Thus, the resulting training set is comprised of seven sampling traces out of the total 34, representing approximately 20% of the data. The duration of these seven traces ranges from 7 to 13 minutes.

Figure 4.15a shows the empirical probability density functions in the training set, where we can identify the behavior described in Section 4.6 for the four datasets previously analyzed. According to the figure, the empirical PDF of user throughput progressively changes as SINR varies. Also, the probability density progression reaffirms the existence of two components. The first component is accentuated at lower SINR values, and the second component becomes more noticeable at higher SINR values (when the first component starts to fade out).

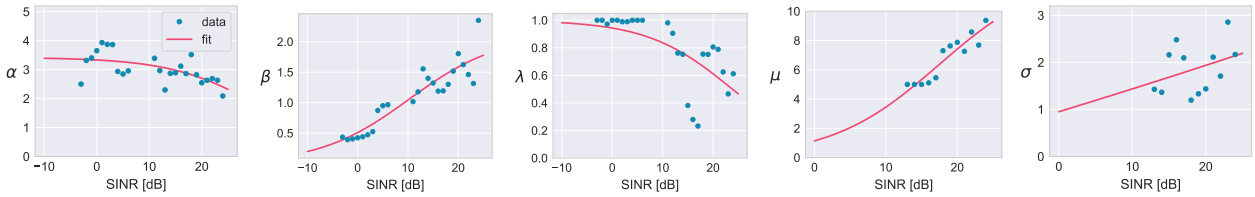


(a) (b)

Figure 4.15: (a) Empirical probability density functions of user throughput according to different SINR values. (b) Box-plot showing the R^2 values obtained by fitting the GGMM to the empirical PDF of user throughput according to every SINR value.

Following our results in Section 4.6, we fit a GGMM to each of the empirical PDFs shown in Figure 4.15a. Figure 4.15b summarizes the coefficient of determination (R^2) obtained by fitting the GGMM to the empirical throughput distributions corresponding to every SINR value. Accordingly, these values confirm the goodness of the fitting model, as the median R^2 score is higher than 0.95.

We filter the fitted models to only those having an R^2 value of more than 0.9. Additionally, for the gamma-component parameters (α and β), we only took into account the fitted GGMMs with parameter $\lambda > 0.1$. Analogous, for the gaussian-component parameters (μ and σ), we only took into account the fitted GGMMs where $(1 - \lambda) > 0.1$. Then, we model the relationship between the filtered parameters and the SINR by applying a generalized logistic regression. The resulting regression models are shown in Figure 4.16.



(a) α parameter. (b) β parameter. (c) λ parameter. (d) μ parameter. (e) σ parameter.

Figure 4.16: Generalized logistic regressions to describe the relationship between the five GGMM parameters and the SINR.

These regression models define the SINR-dependent function F, with which we can estimate the PDF of user throughput for a wide range of SINR values, as shown in Figure 4.17. Next, we can employ F to forecast the user throughput time series in the testing set.

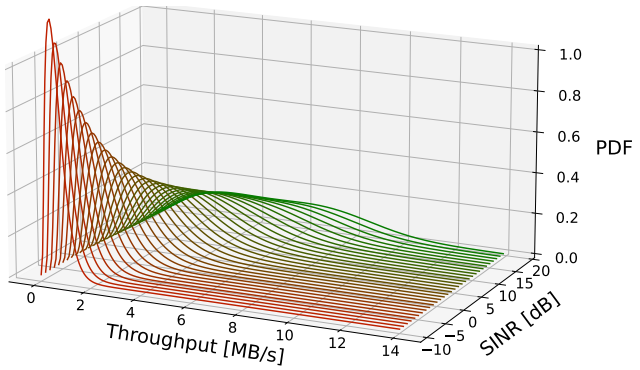


Figure 4.17: Estimated probability density functions of user throughput according to different SINR values.

User throughput can be predicted in different time scales according to the target application. In most cases, throughput is predicted in time windows of one to few seconds [16, 36, 104, 114, 125, 168, 169, 178, 181], and it is rarely predicted in time windows greater than one minute [86, 168].

In order to extend our analysis, we consider the following time scales $w \in \{1, 2, 5, 10, 15\}$ (in seconds). The original traces in the dataset have a time resolution of one second. For the remaining time resolutions, we computed the time series of user throughput and SINR as follows. We took the average user throughput over consecutive time windows of w seconds. For the SINR time series, we took the first SINR value from each time window. Therefore, in each case, the user throughput over a time interval is associated with the instantaneous SINR value at the beginning of the interval.

Thus, we utilized our two proposed throughput prediction methodologies previously described in Section 4.8. With these prediction approaches, we performed a one-step-ahead forecast over the seven user throughput traces in the testing set. We repeat this prediction process for each of the five different time scales. In the following figures, we refer to the prediction based solely on SINR as **SINR** and the prediction based on SINR and previous throughput as **SINR+T**.

As mentioned in Section 4.8.2, our prediction approach based on SINR and previous throughput employs the empirical joint distribution $\overline{f}_{X_n,X_{n-1}}$ that must be previously computed. Nevertheless, we observed that $\overline{f}_{X_n,X_{n-1}}$ significantly varies according to the current SINR value (Y_n). Therefore, we decided to refine the estimation of the joint distribution by using different empirical joint distributions according to the current SINR level. Figure 4.18 shows the empirical joint distributions in the training set using a one-second resolution. This figure illustrates the changes perceived in $\overline{f}_{X_n,X_{n-1}}$ as the current SINR value varies, corroborating the importance of this more refined estimation. As our analysis considers different time scales, we used the training set to perform a preliminary computation of these SINR-dependent empirical joint distributions for each time resolution w, in order to use them by our proposed prediction approach based on SINR and previous throughput (SINR+T).

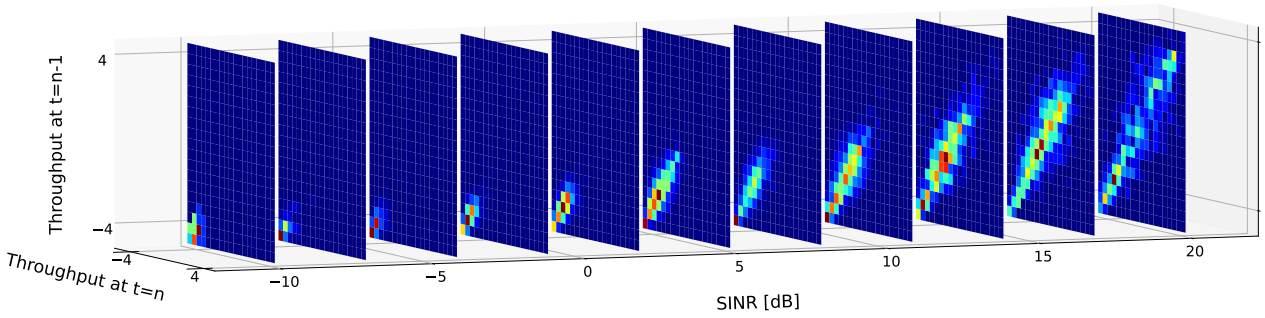


Figure 4.18: Empirical joint distributions computed for different SINR values.

For validation purposes, we consider the Exponentially Weighted Moving Average (EWMA) as a baseline model. Unlike most novel throughput prediction techniques, EWMA can be easily deployed in almost every mobile network scenario, as it does not rely on any other specific information apart from previous throughput values. Indeed, EWMA has been commonly employed as a throughput predictor in real adaptive bitrate applications [169, 179, 13, 68] and it is often considered as a baseline throughput prediction model [86, 105, 169, 136].

Figure 4.19 exemplifies the use of our methodologies (SINR and SINR+T) to predict the user throughput time series in the testing set using a one-second resolution. Analogous, Figure 4.20 shows the prediction of our proposed methodologies over the same trace as Figure 4.19 but with a time resolution of $w = 5[s]$.

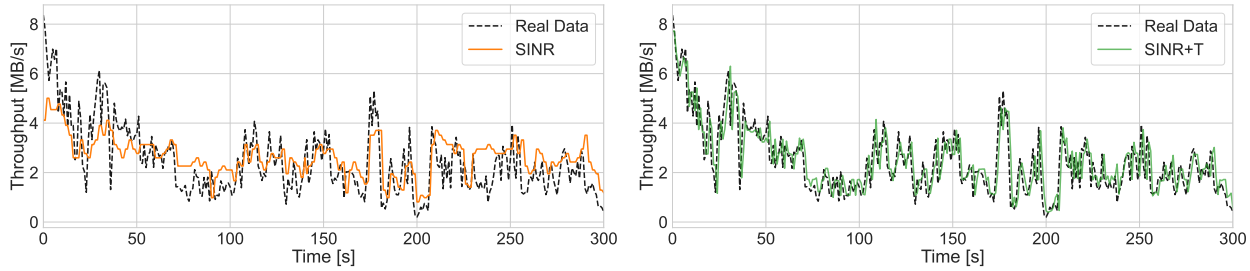


Figure 4.19: Example of throughput prediction using our proposed approaches with a time resolution of $w = 1[s]$.

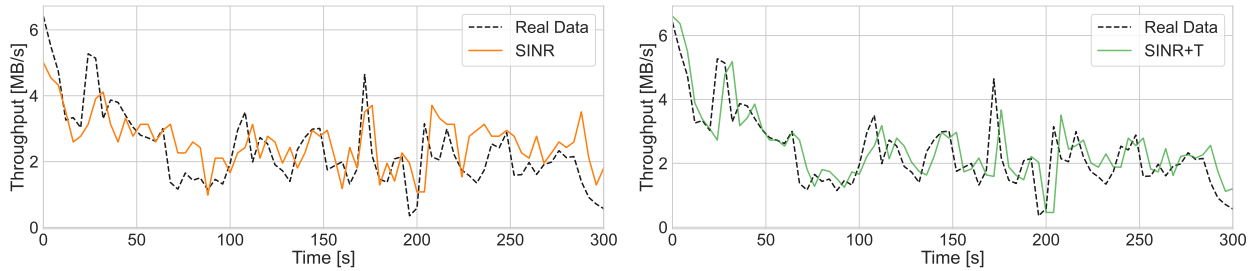
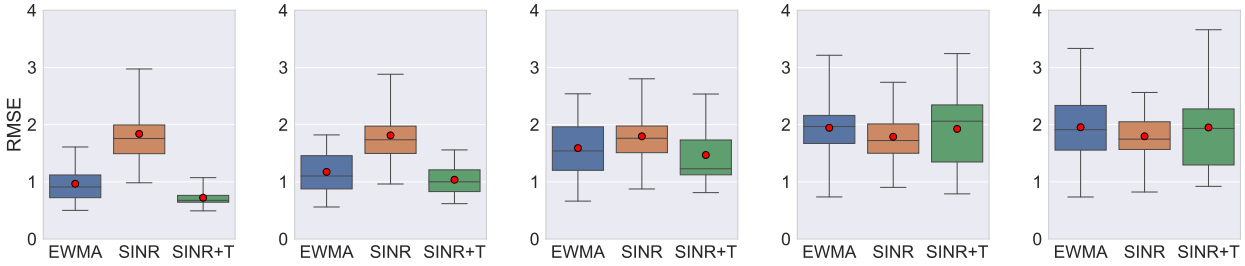


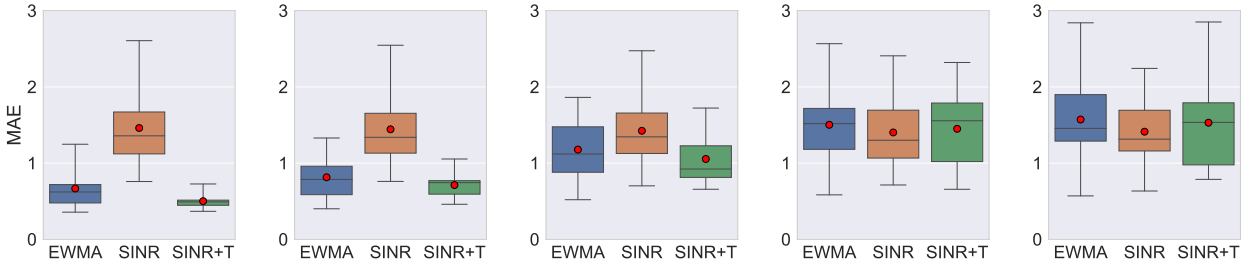
Figure 4.20: Example of throughput prediction using our proposed approaches with a time resolution of $w = 5[s]$.

For each of the five time scales w, we forecast the seven time series using the three selected prediction methods (EWMA, SINR, and SINR+T). Then, we computed the root-mean-squared error (RMSE) and the mean absolute error (MAE) for evaluating each predicted time series. Figures 4.21 and 4.22 summarize the obtained error metrics in this experimental evaluation.



(a) $w = 1[s]$ (b) $w = 2[s]$ (c) $w = 5[s]$ (d) $w = 10[s]$ (e) $w = 15[s]$

Figure 4.21: Box-plots for RMSE of predicted throughput time series using different time resolutions.



(a) $w = 1[s]$ (b) $w = 2[s]$ (c) $w = 5[s]$ (d) $w = 10[s]$ (e) $w = 15[s]$

Figure 4.22: Box-plots for MAE of predicted throughput time series using different time resolutions.

According to our results, the EWMA model exhibits an expected behavior, showing decreasing performance as the time scale increases. Therefore, both RMSE and MAE increase for high values of w, since the most recent values naturally become further away in time from the predicted value.

Interestingly, the prediction based solely on SINR obtained similar error scores over the different time scales, showing even a slight decrease in errors when reaching long time scales for both RMSE and MAE. For short time scales ($w = 1[s]$ and $w = 2[s]$), this prediction method performed worse than the other two models (Figures 4.21a and 4.21b for RMSE, and Figures 4.22a and 4.22b for MAE). However, the stability in the performance of this approach becomes essential for higher values of w, where the performance of the other methods decreases. Indeed, for long time scales ($w = 10[s]$ and $w = 15[s]$), the prediction based solely on SINR achieved the best prediction performance, according to both RMSE (Figures 4.21d and 4.21e) and MAE (Figures 4.22d and 4.22e).

Regarding the prediction based on SINR and previous throughput, it consistently outperforms the EWMA model, according to the employed performance metrics. This is particularly important for short time scales, where this model obtained remarkably lower error values. Similarly to the EWMA, this approach exhibits decreasing performance as the time scale increases, mainly because there is greater prediction uncertainty as the prediction horizon increases.

Therefore, the adoption of our proposed prediction strategies could be beneficial for practical applications since they cover a wide range of scenarios. Indeed, our prediction approach

based solely on SINR is not only useful for scenarios where previous throughput observations are unavailable, but it is also valuable for throughput prediction in higher time scales, where the performance of other methodologies decreases. Additionally, our prediction approach based on SINR and previous throughput is shown to achieve outstanding performance in throughput prediction, especially for short time scales. These results indicate the benefit of predicting user throughput by estimating the mathematical expectation of its probability distribution. Unlike most of the recent research, our approaches do not employ any contextual information but the SINR value, which can be directly obtained from the user equipment. As a result, our prediction methodologies can be deployed and used by applications running on regular commercial mobile devices.

4.10 Summary

In this chapter, we first conducted a comprehensive study on the empirical effect of the SINR on user throughput, paying special attention to the high throughput variability. We proposed and validated the use of an SINR-dependent probabilistic mixture model for estimating the probability distribution of user throughput. This Gamma-Gaussian mixture model was shown to gradually adjust its two components to properly fit the probability density function of user throughput over different SINR values. Then, we extrapolated our SINR-dependent probabilistic mixture model in order to be able to provide an estimation of the probability distribution of user throughput for any particular SINR level. This extensive examination provides a further crucial understanding of the empirical relationship between user throughput and channel quality.

Additionally, we used the probability density estimations given by our SINR-dependent probabilistic mixture model to revisit the problem of throughput prediction in mobile networks. We proposed two prediction strategies, which directly estimate the mathematical expectation of the probability distribution of user throughput. The first prediction approach based solely on SINR was designed as a history-less methodology, and therefore, it can be particularly useful when a connection has not been established yet, and when previous throughput values are no longer valid (e.g., after a vertical handover). Nevertheless, our experimental results exhibit that this approach can also be of great significance for predicting user throughput time series entirely. Indeed, given its performance stability over different time series resolutions, this first approach arises as a valuable alternative, especially for longer time scales. The second prediction approach, based on SINR and previous throughput, properly captured the effect of prior throughput values by calculating the conditional expectation of throughput. This prediction approach consistently outperformed the EWMA baseline model, and therefore, it arises as a practical user throughput prediction choice, especially for shorter time scales. Consequently, the adoption of our proposed methodologies can be beneficial for practical applications since they (*i*) cover a wide range of application scenarios, (*ii*) are easily explainable as estimations of the mathematical expectation of user throughput, and (*iii*) can be deployed on regular mobile devices as they only rely on the observed SINR.

Chapter 5

Comments on Wireless Channel Quality Measurements

5.1 Introduction

While studying the anticipatory networking problem related to mobile QoS (Chapter 3 and Chapter 4), we persistently dealt with different signal quality metrics for mobile networks, such as the Received Signal Strength Indicator (RSSI), the Reference Signal Received Power (RSRP), and the Signal-to-Interference-Plus-Noise Ratio (SINR). Since these values are commonly expressed in logarithmic units, they must be carefully handled. Nevertheless, when analyzing the related literature, we found some recurrent issues related to the mishandling of log-scaled signal quality metrics. Thus, in this chapter, we discuss and propose solutions to enhance the handling of signal quality measurements, with a special focus on Mobile Crowdsourcing scenarios.

During the last decade, many research studies have made use of Mobile Crowdsourcing methods to analyze the performance and quality of service (QoS) in mobile environments. These studies usually obtain different QoS indicators together with some environmental data, such as timestamps, location coordinates, and cell identifiers, to describe wireless network behavior for a given geographical area. Among all the collected network information, the received signal strength indicator is included in most Mobile Crowdsourcing analyses. This recurrent consideration of signal strength is in part because it is very easy to obtain from end-user mobile devices [157], but mostly because of its influence on the overall QoS in wireless networks, which is reflected in the impact produced by signal strength variations in network performance measurements [59, 120, 157, 150]. Moreover, analyses over signal strength data are not only interesting for academic research but also for mobile analytics companies such as *OpenSignal* and *Tutela*, and for mobile network operators for radio network planning [123, 128] and for performing coverage analysis in cellular networks [43].

A common methodology to summarize the received signal strength inside a specific area is to aggregate all the individual measurements into one representative value that characterizes the signal strength inside the location area [161, 174, 93, 147, 157, 146, 99, 85, 107, 165, 173,

80].

The first question about getting a representative signal strength value from the aggregation of several individual measurements is to identify what real value we actually want to represent and estimate. In this work, we consider the formal definition of the expected value of signal strength as the target value to be estimated. The expected value is a measure of central tendency, i.e., a value for which the results will tend to. Intuitively, it is the theoretical mean value of a random variable over a large number of experiments, and it is commonly used to summarize all the information about a random variable in a single numerical value.

In Mobile Crowdsourcing scenarios, signal strength samples are taken by real end-user devices with custom measurement apps. This leads to important sources of error to take into account when aggregating values inside an area:

1. Measurements are not uniformly distributed in the area, as they are defined by human mobility patterns [113].
2. The number of measurements in small areas (e.g., the coverage area of a single cell) could not be high enough to be considered representative enough [157].
3. The measurements present accuracy errors in both signal strength values and geographic coordinates [157].

Hence, some commonly used methods to characterize signal strength could not necessarily return a good estimation of the expected value of signal strength, since they do not consider the aforementioned sources of error, which are present in most Mobile Crowdsourcing signal strength data.

In this chapter, we present two clear contributions to the related literature. Firstly, we present a formal analysis of how signal strength values must be handled to avoid some common pitfalls in using log-scaled signal strength. Secondly, we present a novel Aggregation Method Based On Interpolation of signal strength (ABOI method). Our proposed method obtained consistently lower RMSE values than other commonly used methods for estimating the expected value of signal strength over an area, in both simulated and real scenarios. Consequently, the ABOI method is demonstrated to be more robust against the existing difficulties of real-world measurements.

5.2 Background

Due to its large impact on the overall QoS of wireless networks, many research works have focused on characterizing the received signal strength for an area of particular interest. These analyses frequently utilized all the individual signal strength samples taken by each mobile device sensing the network.

In this way, some works aggregated several signal strength samples inside the same area into a unique representative signal strength value to predict user availability [174], measure

the effect of weather conditions in the received signal strength [93, 147], analyze network performance [146], measure the impact on the signal strength of indoor-outdoor context [99] and find correlations between signal strength and other QoS indicators for mobile networks as network congestion [107], throughput [161], and TCP goodput and latency inside the same geographic area [157]. In this work, we propose an aggregation method that better estimates the expected value of signal strength than the methods used in the related studies mentioned above, especially when using measurements taken in Mobile Crowdsourcing contexts. Thus, the results and conclusions of these works can be refined by using our proposed model.

It is important to notice that there are other research works that also aggregated several signal strength samples, although not to find a representative signal strength value. Some of these works used signal aggregation to perform base transceiver station (BTS) localization [84] or to estimate user location [85, 116], mostly based on the RADAR system [12]. These studies are not related to the problem we are referring to in this chapter, which is to aggregate signal strength samples into a representative value to estimate the mathematical expectation of the signal strength.

There are some works that developed high-resolution coverage maps from Mobile Crowdsourcing signal strength measurements. These maps were created by plotting each empirical sample on the map [39, 97] or by interpolating the signal strength in several uniformly distributed points inside the area of interest using linear interpolation [162], using variations of the Kriging method [109, 40] or by using Gaussian processes that consider prior knowledge about theoretical path loss models [149]. These coverage maps are useful for tasks that require highly detailed maps, but when analyzing signal coverage in greater areas, the effectiveness of their fine-grained visualizations will decrease as the resolution of the maps decreases. Positioning all individual samples will greatly increase the clutter in the visualizations, defeating the purpose of providing useful information for the measured areas. Therefore, for these cases, it is also important to consider the aggregation of signal strength samples, to be able to generalize their results to maps with lower resolution, describing the signal strength in larger areas by only one representative value. Consequently, the method for signal strength aggregation proposed in this chapter could also be useful for these works.

Some works that employed Mobile Crowdsourcing data discussed the problem of not having uniformly distributed samples over the measured area [109, 149] and how the spatial distribution of the samples matches population patterns [77]. This is important, since some studies that used simulated Mobile Crowdsourcing data to evaluate their methods, implicitly assumed uniform spatial distribution of the samples (as shown in Section 5.4.1). Uniform distribution is not a realistic measurement scenario, especially when samples are taken by real end-user mobile devices. For better reliability, we consider in our experiments both uniform distribution of signal strength samples and distribution based on social network theory [113], which is closer to the spatial distribution present in real Mobile Crowdsourcing measurements.

5.3 Common Pitfalls Using Log-scaled Signal Strength

The use of log-scaled signal strength values is a widespread methodology for analyzing radio frequency measurements. Signal amplitude could vary very widely, and therefore, it could be difficult to analyze and understand the relationships among different values in the linear watt scale. Hence, using log-scale enhances signal strength visualizations by improving the display range of axes. Using log-scaled signal strength values is also attractive since it can lead to data compression, requiring fewer bits of information [142].

Log-scaled signal strength values could be used in dBm units (decibels with reference to one milliwatt) or in Arbitrary Strength Units (ASU), since ASU values are linearly proportional to the received signal strength in dBm, and consequently, they are also logarithmic values.

Before using and manipulating dBm values, it is important to analyze the origin of dBm from a physical point of view, regarding *dimensional analysis*. Power is a derived quantity that can be expressed in terms of fundamental units (time, length, and mass). In fact, power values must have dimension ML^2T^{-3}. The International System of Units (SI) describes the watt (symbol: W) as a unit of power, defined as a derived unit in terms of base units, where $1\ \mathrm{W} = 1\ \mathrm{kg} \cdot \mathrm{m}^2 \cdot \mathrm{s}^{-3}$. In addition, the prefix *Milli-* (symbol m) has been part of the SI since 1960 and only denotes a factor of 10^{-3}. This prefix never changes the units, and therefore, values expressed in milliwatt (mW) are actually being expressed in watt (W) units.

If we consider a power value P_{mW} expressed in milliwatts, its corresponding value in dBm is formulated as

$$P_{dBm} = 10 \log_{10} \left(\frac{P_{mW}}{1\ \mathrm{mW}} \right) \tag{5.1}$$

It is clear to see that the right side in Equation (5.1) is dimensionless, since the mW dimension is canceled in $\frac{P_{mW}}{1\ \mathrm{mW}}$. Then, the parameter inside the logarithm function in Equation (5.1) is a dimensionless number, and therefore, P_{dBm} is also dimensionless. This fact is essential because, even when P_{mW} has dimensions, there is no physical sense for P_{dBm} to have it. Thus, it is a mistake to consider dBm as a power unit, since it does not meet the dimension of power quantities (ML^2T^{-3}).

The above is documented by Sonin [155]: "*Products, ratios, powers, and exponential and other functions such as trigonometric functions and* ***logarithms*** *are defined for numbers, but* ***have no physical correspondence in operations involving actual physical quantities***".

Furthermore, the mere fact that P_{dBm} is defined as a logarithmic function implies that dBm is a dimensionless quantity. In fact, we consider the formal definition of $10 \log_{10}(x)$:

$$10 \log_{10}(x) = \frac{10}{\ln(10)} \int_{t=1}^{t=x} \frac{dt}{t}$$

The integral $\int_{t=1}^{t=x} \frac{dt}{t}$ corresponds to the sum of an infinite number of terms $\frac{dt}{t}$. All of these terms are dimensionless, and therefore, the whole expression $10\log_{10}(x)$ will always be dimensionless. Then, further interpretations of the dimensionality of dBm should not be accepted since dBm values are intrinsically dimensionless.

This non-coherency in dimensionality between dBm values and power quantities (ML^2T^{-3}) can be evidenced by the knowledge of the use of dimensional formulas in changing units [22], where there is no possible transformation to change from W to dBm consistently. The consequence of the aforementioned is that dBm values do not meet *Bridgman's principle of absolute significance of relative magnitude* (Lemma 1), which is essential to all the systems of measurement in scientific use [21].

Lemma 1 dBm values **do not** meet *Bridgman's principle of absolute significance of relative magnitude.*

Proof. Let SQ be a secondary quantity described by

$$SQ = f(\alpha, \beta, \gamma, ...),$$

where $\alpha,\ \beta,\ \gamma, ...$ are primary quantities and f is the function that combines them.

SQ satisfies Bridgman's principle of absolute significance of relative magnitude if

$$\frac{f(\alpha_1, \beta_1, \gamma_1, ...)}{f(\alpha_2, \beta_2, \gamma_2, ...)} = \frac{f(x\alpha_1, y\beta_1, z\gamma_1, ...)}{f(x\alpha_2, y\beta_2, z\gamma_2, ...)} \tag{5.2}$$

holds for all values of $\alpha_1,\ \beta_1,\ \gamma_1, ...,\ \alpha_2,\ \beta_2,\ \gamma_2, ...$ and for all coefficients $x,\ y,\ z, ...$ [21].

As stated in Equation (5.1), dBm can be described as a function of primary quantities:

$$f_{dBm}(\alpha, \beta, \gamma) = 10\log_{10}\left(\frac{\alpha[kg]\cdot\beta[m^2]\cdot\gamma[s^{-3}]}{10^{-3}\cdot 1[kg]\cdot 1[m^2]\cdot 1[s^{-3}]}\right) \tag{5.3}$$

Proceeding by contradiction, assume that dBm values do meet Bridgman's principle of absolute significance of relative magnitude. Then, the Equation (5.2) should hold for f_{dBm} in Equation (5.3) and for all values of $(\alpha_1, \beta_1, \gamma_1)$, $(\alpha_2, \beta_2, \gamma_2)$ and (x, y, z). In particular, it should hold for the following values:

$$\begin{array}{lll} \alpha_1 = 10^{-9} & \beta_1 = 1 & \gamma_1 = 1 \\ \alpha_2 = 10^{-8} & \beta_2 = 1 & \gamma_2 = 1 \\ x = 10 & y = 1 & z = 1 \end{array}$$

By replacing these values in the left side of Equation (5.2):

$$\frac{f_{dBm}(\alpha_1, \beta_1, \gamma_1)}{f_{dBm}(\alpha_2, \beta_2, \gamma_2)} = \frac{-60}{-50} = 1.2$$

And by replacing these values in the right side of Equation (5.2):

$$\frac{f_{dBm}(x\alpha_1, y\beta_1, z\gamma_1)}{f_{dBm}(x\alpha_2, y\beta_2, z\gamma_2)} = \frac{-50}{-40} = 1.25$$

Both sides of the equation are not equal, which is a contradiction. Then, since the relationship is not fulfilled for all values, we conclude that dBm values (represented as f_{dBm}) do not satisfy Bridgman's principle of absolute significance of relative magnitude.

□

As a direct consequence of Lemma 1, equations involving dBm units are considered as **not physically relevant** [155].

Meeting Bridgman's principle is, according to Percy Bridgman [21], essential to all the systems of measurement in scientific use. This principle is fundamental to guarantee that the selection of a different unit of measurement will not affect the outcomes of any experiment. Therefore, as dBm values do not meet Bridgman's principle, some numerical relationships among power values in W do not remain true when using dBm. That is, the outcomes of scientific experiments can be affected if using dBm values instead of watt values. This should not be allowed in scientific research, as nature is indifferent to the arbitrary choices we make when we pick base units. Indeed, as Sonin [155] precisely stated: "*Nature is indifferent to the arbitrary choices we make when we pick base units. We are interested, therefore, only in numerical relationships that remain true independent of base unit size.* ". However, dBm values do not respect this, as evidenced in the following straightforward example:

$$1 \text{ mW} + 1 \text{ mW} = 2 \text{ mW}$$

If we transform all values from mW to dBm using Equation (5.1), we have

$$0 \text{ dBm} + 0 \text{ dBm} = 3.0102999566 \text{ dBm}$$

This is, of course, wrong and contradictory, and it exemplifies that if we wrongly attempt to perform addition of dBm values, we will reach erroneous conclusions such as 1 mW = 2 mW. Accordingly, to be coherent with the dimensional analysis and with the mathematical basis, all mathematical operations involving signal strength must use linear watt values.

Some early research studies (before the 1980s) purposely included these wrong methodologies in their analyses. However, when applying mathematical operations to log-scaled signal values, they had a clear understanding of the definition and implications of using logarithmic power values. As they stated, they performed these methodologies to compare how different

their results would be if using log-scaled signal values [106, 55], or to explore the *"attractiveness of the logarithm of power"* such as its contribution to compression of data requiring fewer bits of information [142] (what may have been a genuine concern at that time). Nevertheless, we did not find any discussion or argument on why to use log-scaled signal values in more recent papers. Indeed, many of these papers manipulated dBm values without mentioning the correspondence between dBm and watt values [52, 84, 40, 28, 107, 99, 33, 88, 153, 116, 73], and moreover, some of them manipulated signal strength values without reporting the unit of measurement employed [77, 85, 41, 146, 74]. Many of the papers that followed these wrong methodologies got log-scaled signal strength measurements directly from mobile operating systems (Android or iOS) [28, 52, 161, 84, 147, 40, 99, 109, 107, 153, 39, 146, 33, 157]. Therefore, it is plausible that they just used and manipulated the data returned by the systems without a thorough analysis of the unit of the collected signal strength values.

It is important to understand that applying mathematical operations with log-scaled signal strength values involves wrong models and interpretations of reality, and therefore, leads to wrong conclusions. Nevertheless, related research works have frequently made these mistakes. Many of these papers have been published during the last few years, evidencing that the misuse of log-scaled signal strength values is a real problem within the mobile computing community nowadays. These methodologies must be avoided, even when they have been constantly used in the past, since their habitual use does not validate their contradiction with some basic principles of scientific analysis.

The following subsections describe some of these common but misinterpreted practices.

5.3.1 Averaging Signal Strength

The average of signal strength measurements taken in similar temporal space conditions has been widely employed. For instance, the arithmetic mean of measurements taken in a single point can be used to reduce measurement variance, since every signal strength sample is assumed to be contaminated with unrelated additive noise. Moreover, the arithmetic mean of measurements inside the same geographic area can be used to obtain a representative value of signal strength, getting an estimation of the mathematical expectation of signal strength in the area (as shown in Section 5.4.1).

The arithmetic mean involves taking the sum of samples; however, we already stated the lack of physical relevance of the addition (and any other equation) involving log-scaled signal values. Consequently, the arithmetic mean of log-scaled signal values can not either be considered physically relevant. This bad practice implies, in most cases, a distortion of real signal strength behavior [55, 142], as shown in the following simple but explanatory example. Let a be a vector of signal strength values in dBm units:

$$a = \begin{bmatrix} -45 \text{ dBm} & -55 \text{ dBm} \end{bmatrix} \tag{5.4}$$

The arithmetic mean of samples in a (in linear scale) is 1.74e$-$5 mW, which is equal to -47.6 dBm. Instead, the arithmetic mean of log-scaled samples in a is -50 dBm, with an

error of 2.4 dB from the real value introduced by this misleading methodology. Although these errors may seem small in some cases, they should not be underestimated due to the impact of signal strength fluctuations on other important network performance metrics [59, 120, 157, 150]. Differences around 5 dB in signal strength could imply, in some cases, an increase of 100% in packet loss rate and round-trip time of a connection over the mobile network [120].

Despite the aforementioned, there are works in which several signal strength samples were aggregated by performing a log averaging process, taking the arithmetic mean of dBm or ASU measurements, misunderstanding signal strength real behavior. Some works that used this incorrect methodology are listed below:

- (2017) Sabu et al. [147] conducted a correlation study between signal strength and rainfall intensity in an area of interest, where logarithmic ASU values were aggregated by taking the arithmetic mean. As a result, the authors concluded that the drop in signal strength during rainfall was not as significant as expected by the theoretical hypothesis.
- (2018) In the data exploration section provided by Sung et al. [161], an area of interest was divided into smaller square areas. For each square area, signal strength was reported by taking the mean of several logarithmic ASU measurements. As a result, a weak geographical correlation between signal strength and throughput was found.
- (2015) In the research work of Marina et al. [99], signal strength samples in dBm units were divided according to their context (indoor or outdoor), and then aggregated by taking the arithmetic mean. Then, the authors analyzed the great impact of user context (indoor or outdoor) on the received signal strength.
- (2013) Sonntag et al. [157] created signal strength coverage maps by taking the arithmetic mean of values represented as percentages. The percentage values are calculated linearly from logarithmic signal strength, so they are also logarithmic values. The authors concluded that coverage maps created from crowdsourced signal strength were not very good for describing the actual transport quality.

In addition to the above, there are other research papers that also followed the incorrect methodology of calculating the arithmetic mean of logarithmic signal strength values [28, 52, 84, 153, 146, 33, 116, 173].

The only recorded case in radio frequency analysis where directly average log-scaled measurements using the arithmetic mean is admitted, is when dealing with samples of a *repetitive* or *continuous wave* signal. For this case, averaging log-scaled values is equal to log-scaling the average of linear values [111]. However, this is not the case of the signals we are referring to in this work. Therefore, the introduced error by averaging log-scaled values depends on the statistics of the power estimates being averaged [142, 106]. Consequently, considering that log-scaling the linear average will be equal to the average of the log-scaled values, and that the introduced error can be ignored, are incorrect assumptions.

5.3.2 Comparing Signal Strength

When comparing two signal strength values, e.g., to calculate a prediction error, it is essential to properly measure the difference between the values. For example, if the signal strengths we want to compare are −50 dBm and −45 dBm, then it is correct to say that the values differ in 5 dB, which represents the relation between both signal strengths. It is also correct to say that the difference between these values is 2.162e−5 mW (the difference in linear scale) or, equivalently, −46.65 dBm, although this latter form in dBm units can be quite confusing. However, it is a big mistake to say that the difference between −50 dBm and −45 dBm is 5 dBm, since this equals 3.162 mW, which is several orders of magnitude greater than the real difference shown before. Despite the above, some works performed signal strength comparisons using this last incorrect methodology [99, 52, 153]. This misunderstanding is evidenced in sentences such as *"median error is 6 dBm"* [153], *"differing by more than 15 dBm"* [99], *"the real signal strength is 2.5 dBm stronger* [33] or *"the errors are 10 dBm, 7 dBm, and 6 dBm, respectively"* [52]. These error levels are not coherent with the data used, where there are practically no values greater than −50 dBm.

In addition, when comparing different signal strength prediction errors to decide which error is lower, it is essential to compare them using absolute errors (in watt or dBm scale) rather than relative errors (in dB). The comparison of prediction errors using relative values, could lead to misunderstandings illustrated in the following simple but explanatory example. An error of +4 dB in predicting −70 dBm could be considered lower than an error of +5 dB in predicting −110 dBm, holding that 4 dB $<$ 5 dB. However, if we analyze these errors as linear absolute errors, we find that the first error is actually more than $10,000$ times greater than the second one, giving an absolutely opposite view than the one obtained by analyzing relative dB values. An example of the use of this incorrect methodology is described in the following:

- (2019) Alimpertis et al. [3] proposed a new method based on machine learning to perform signal strength prediction, i.e., given a set of signal strength measurements in an area, estimate signal strength values in other singular points. They claimed that their method consistently obtains lower prediction errors than related state-of-art algorithms. Nevertheless, it can be shown that the comparison of their errors leads to inconclusive results. Using the values shown in Table 4 from their research study [3], for cell ID $x204$, their method obtains an average error of 2.3 dB, outperforming Ordinary Kriging (OK) and Ordinary Kriging Detrending method (OKD) which obtains average errors of 3.85 dB and 2.99 dB respectively. However, if we consider the case in which their method's error is +2.3 dB, OK's error is −3.85 dB, and OKD's error is −2.99 dB, all in relation to the expectation of the signal strength of cellID $x204$ (−96 dBm), we have that OK method has an error 26% lower than their method's error, and OKD method has an error 30% lower than their method's error (using linear watt scale). In that case (a possible case given the prediction errors stated in the paper), their method actually gets worse results than related state-of-art algorithms.

Another common task is to summarize several prediction errors for $(real, forecast)$ pairs of signal strength values. This is performed by calculating different measures of prediction

accuracy such as MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error), or MASE (Mean Absolute Scaled Error). Nevertheless, many of these works failed in estimating their prediction accuracy by using a logarithmic scale for prediction errors, and summarizing them by applying mean-based aggregation such as MAPE [40, 109], MSE (Mean Squared Error) [149], or RMSE (Root Mean Squared Error) [3]. Then, they added a source of error at applying mean functions to log-scaled values, as mentioned in Section 5.3.1. In fact, as it is well known that the average of the logs will always be less than or equal to the log of the average [55], applying mean-based error measures to log-scaled errors, will imply an underestimation of real errors.

Signal strength samples in linear scale should be preferred for estimating the error between two signal strength values and for summarizing several errors in an accuracy measure. However, this does not prevent these results from being used later in log-scale if desired (for example, for visualization).

5.4 Signal Strength Aggregation

As mentioned in Section 5.1, we consider each aggregated value as an estimation of the mathematical expectation of signal strength:

For an area A, we consider the function $P(\vec{p})$, which represents the signal strength in function of the position $\vec{p}$. Thus, a representative signal strength value for A, obtained from the aggregation of individual measurements, will try to be as close as possible to the mathematical expectation $\mathbb{E}(P(X))$, where X is a uniformly distributed random variable of position in A.

We define $\overline{P}_A$ as the division of the integral of the function P in A, and the total area A, which is equal to $\mathbb{E}(P(X))$ as shown below:

$$\mathbb{E}(P(X)) = \int_{\Omega} P(\omega) f_X(\omega) d\omega = \frac{\iint_A P(\vec{p}) dA}{\iint_A dA} =: \overline{P}_A \tag{5.5}$$

where Ω denotes the set of all positions ω in A, f_X is the probability density function of X, which is a constant equal to $\frac{1}{\iint_A dA}$, and $\vec{p}$ denotes the position in A.

Considering a discretization of the space, the mathematical expectation in Equation (5.5) can be approximated by Riemann sums:

$$\overline{P}_A \approx \frac{\sum_{i=1}^{m} P(x_i) \Delta A_i}{\sum_{i=1}^{m} \Delta A_i} \tag{5.6}$$

and, when the discretization is such that all the points are equispaced, it follows that $\Delta A_i = \Delta A$ is constant for all $i = 1, ..., m$. The approximation becomes better as ΔA gets smaller, and consequently, the number of points (denoted by $m_{(\Delta A)}$) gets larger. Accordingly, the approximation by Riemann sums corresponds to the arithmetic mean and fulfills that:

$$\lim_{\Delta A \to 0} \frac{1}{m_{(\Delta A)}} \sum_{i=1}^{m_{(\Delta A)}} P(x_i) = \overline{P}_A \tag{5.7}$$

Thus, as the equispaced discretization becomes finer, the better the approximation of $\overline{P}_A$. The main drawback of this approximation method is the need to know the value of P in several equispaced positions over A to obtain an accurate estimation.

Another strategy to approximate the expected value relies on considering that the positions of the measurements x_i, $i = 1, ..., m$, are given by independent uniform random variables over the area A. Then, we can use the Monte Carlo method to approximate $\overline{P}_A$ and, by the law of large numbers, we have that

$$\lim_{m \to \infty} \frac{1}{m} \sum_{i=1}^{m} P(x_i) = \overline{P}_A \tag{5.8}$$

The expression in Equation (5.8) is equivalent to Equation (5.7), and corresponds to the arithmetic mean. This is the value that past works referred to as the "local mean signal strength", used to summarize signal strength in areas of a few meters (up to 40 wavelengths) [165, 173, 80]. In fact, local mean signal strength *"is obtained by averaging a large number of individual RF measurements taken in a local neighborhood"* [165]. Thus, related studies that estimate local mean signal strength are actually estimating the mathematical expectation of signal strength.

In the following, we discuss algorithms for estimating $\overline{P}_A$, using data from Mobile Crowdsourcing apps. First, we consider two commonly used performance metrics as aggregation methods: the arithmetic mean and the median value. In addition, we propose a novel method based on the interpolation of signal strength values.

5.4.1 Arithmetic Mean

A simple method to summarize signal strength measurements is to take the arithmetic mean $\overline{x}_A$ of all the samples in area A, as commonly used in Mobile Crowdsourcing contexts [99, 147, 161, 77, 157, 165]:

$$\overline{x}_A = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{5.9}$$

This is a good first approach to estimate $\overline{P}_A$, based on the fact that, if the samples

are independent and uniformly distributed in area A, then $\overline{x}_A$ is an example of the Monte Carlo method, which assures that $\overline{x}_A$ converges to $\overline{P}_A$ when $n \to \infty$, as shown in Equation (5.8). Indeed, the law of large numbers and the Monte Carlo method could apparently justify the use of the arithmetic mean as an estimator of the mathematical expectation of signal strength in area A. Nevertheless, research studies that used the arithmetic mean over signal strength samples did not look over the fulfillment of the hypothesis required by the Monte Carlo method. Firstly, in a realistic Mobile Crowdsourcing scenario, the number of signal strength samples could be low for small areas. Secondly, crowdsourced signal strength measurements would not be sampled uniformly on area A, as their positions are determined by human mobility patterns. Accordingly, there is no real mathematical foundation for using the arithmetic mean to estimate the expected value of signal strength in these measurement contexts.

Estimating $\overline{P}_A$ by taking the arithmetic mean of Mobile Crowdsourcing data, is based on a *convenience sampling* process that only considers measurements from locations that are readily available or easy to reach. In this case, readily available locations are directly defined by the mobility of test users. The use of this sampling method is well known to be likely to have biased results, because selecting cases based on their availability does not allow a generalization to the total population [54]. In our case, this means that the estimation of $\overline{P}_A$ will be biased by the locations in area A where test users took measurements.

As mentioned in Section 5.3.1, the arithmetic mean of signal samples should be taken over linear values to avoid induced bias due to incorrect methodologies. In addition to these physical and mathematical formalities, the importance of using linear values to better estimate the expected value of signal strength has also been stated in the past: *"In terms of accuracy, the preferred method for estimating the local mean signal strength at a specific point is to average (in **watts**) a **large number** of individual RF measurements"* [165]. Nevertheless, some works use the wrong methodology by explicitly applying the arithmetic mean over signal strength in logarithmic scale [147, 161, 99, 157] as stated in Section 5.3.1.

Due to the above, in this work, we consider only the correct mean of samples in linear scale.

5.4.2 Median Value

Another method to summarize several signal strength samples is by choosing the median value that separates the higher half from the lower half of all the measurements [93, 61, 127, 66]. The idea behind using median value to aggregate signal strength samples is that it is not skewed so much by a small proportion of extremely large or small values, which is a common situation in this case study, because, as shown in Section 5.3, signal amplitude could vary very widely among measurements.

Furthermore, since the logarithm is a strictly increasing function, the median value has the advantage that it will be the same value if selected from signal strength values in linear (watt) or logarithmic (dBm) scale.

In cases such as the Gaussian distribution, the median value is a good estimator for the

mathematical expectation, since the latter naturally separates the higher from the lower half of possible values. Nevertheless, this assumption is not very likely to be true for Mobile Crowdsourcing contexts, where the signal strength distribution depends on the positions of base transceiver stations with respect to the area of interest.

In addition to the above, the median value method also induces a bias due to the *convenience sampling* of measurements. Therefore, the median value is not expected to perform well in estimating the mathematical expectation of signal strength in Mobile Crowdsourcing scenarios, as there is no mathematical foundation for its use. However, due to its wide use in the literature, it is important to consider the median value as a baseline of our study to quantify the error it can reach estimating $\overline{P}_A$.

5.4.3 Our Proposal: Average Based on Interpolation (ABOI)

As shown in Section 5.3.1, most Mobile Crowdsourcing scenarios do not fulfill the required hypotheses to employ the arithmetic mean as an estimator of the mathematical expectation of signal strength (hypotheses for Monte Carlo integration). Therefore, we wanted to design a more robust method to estimate the expected value from signal strength measurements, without requiring the samples to be independent and uniformly distributed.

For our proposed method, we return to the idea of estimating the mathematical expectation of signal strength using Riemann sums, according to Equation (5.6). As mentioned in Section 5.1, to obtain better approximations to the real $\overline{P}_A$ by using Riemann sums, we need an equispaced grid of signal strength values as fine-grained as possible. However, as stated in Section 5.2, it is not possible to ensure a high number of measurements and uniform spatial distribution in most Mobile Crowdsourcing scenarios. To solve these problems, we use the available measurements to interpolate the signal strength in a fine-grained grid, obtaining equispaced data and increasing the number of available samples. Thus, to estimate $\overline{P}_A$, we take the arithmetic mean of all values in the fine-grained grid G (in watt) as shown in Equation (5.7), avoiding the difficulties of nonuniform spatial distribution and a low number of measurements. Consequently, to obtain a good estimation of the mathematical expectation, we need to establish the conditions on the signal strength measurements that guarantee proper interpolation. As our proposed method is an average based on interpolation, we will refer to it as ABOI.

Although there are many interpolation methods, it is out of the scope of this work to discuss the advantages and disadvantages of each one. For the interpolation step in the ABOI method, we use one of the simplest and most commonly employed interpolations methods in signal strength analysis: the Ordinary Kriging (OK) algorithm. Nevertheless, the ABOI method could be improved by using a more complex and accurate interpolation algorithm.

To estimate the value of signal strength at a position x_0 on the grid, the OK algorithm takes a linear combination of its neighbors:

$$P^*(x_0) = \sum_{i=1}^{n} \omega_i P(x_i) \tag{5.10}$$

where x_i represents all the neighbors of x_0, and ω_i is the corresponding weight of each neighbor. In general, ω_i is proportional to the distance between x_0 and x_i.

It is important to notice that many authors wrongly used this method with dBm values or simply do not make explicit the scale used [74, 73, 3, 52]. As mentioned in Section 5.3, we emphasize that this algorithm should be used on the linear power scale, since it involves algebraic operations, such as addition and weighting.

5.5 Mathematical Foundations for the Use of the ABOI Method

In this Section, we present the mathematical foundations for using the ABOI method to estimate the expected value $\overline{P}_A$ of signal strength measurements inside an area A of interest. Firstly, we announce Theorem 2, establishing the conditions under which the error of the estimation of $\overline{P}_A$ provided by the ABOI method can be smaller than ε. The hypotheses required for this result are shown to be consistent with realistic Mobile Crowdsourcing scenarios, contrarily to the case of arithmetic mean, as stated in Section 5.4.1. Lastly, we demonstrate that the ABOI method is an improvement on the arithmetic mean in estimating $\overline{P}_A$. That is, signal strength measurements that are favorable for the arithmetic mean (that do fulfill Monte Carlo integration hypotheses) are still favorable for the ABOI method. However, favorable cases for the ABOI method can be very disadvantageous for the arithmetic mean.

5.5.1 ABOI Theorem

ABOI Theorem (Theorem 2) specifies the conditions under which the error of ABOI's estimation can be smaller than ε, providing a proper approximation of the expected value of signal strength. For that purpose, some important definitions need to be stated first.

Let $N = \{x_1, x_2, ..., x_n\}$ be the set of positions of the initial n signal strength measurements taken inside a rectangle area $A = [a_1, b_1] \times [a_2, b_2]$. Analogously, let M be the set of positions of the m points equispaced over A on which the ABOI method interpolates signal strength. Sets N and M are exemplified in Figure 5.1.

Let us consider the following definition of the fill-distance:

$$h_N := \sup_{x \in A} \min_{x_i \in N} ||x - x_i|| \tag{5.11}$$

The value h_N indicates the largest distance between each position in A and its nearest neighbor in N (original measurements).

Let us call ABOI(N,M) the return value of the ABOI method after using the n original measurements to interpolate signal strength (using OK) on grid M, and computing the

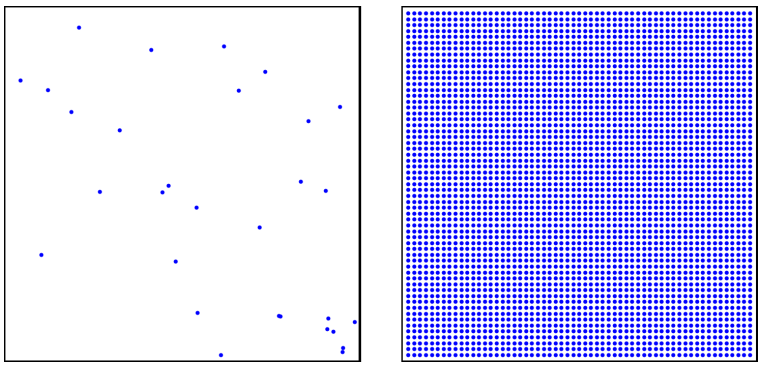


Figure 5.1: Example of set N with $n = 30$ positions of initial measurements (left), and set M with $m = 3481$ equispaced positions over A where to interpolate signal strength (right).

arithmetic mean of the m interpolated watt values.

Theorem 2 Given that power measurements $P(\cdot)$ can be modeled by Gaussian Processes ($\star$). Let $\varepsilon > 0$ be the desired error level for the estimation of the expected value provided by ABOI(N)(M). Let A be a rectangle area where to estimate the mathematical expectation of signal strength. If the n initial measurements are such that h_N is small enough ($\star\star$), and selecting M as a fine-grained enough grid over A ($\star\star\star$), then

$$\mathbb{E}\Big[\big|\mathbb{E}_A(P(X)) - \text{ABOI(N,M)}\big|\Big] \leq \varepsilon$$

that is, the expected value of the error between the mathematical expectation of signal strength over A and the estimation provided by the ABOI method is smaller than the given ε.

Proof. Let P_M be the arithmetic mean of real signal strength values on each position in M. These m values are unknown when applying the ABOI method. Nonetheless, P_M will be helpful to bound the expected value of the estimation error. Indeed, we can bound the estimation error of the ABOI method as follows:

$$\begin{aligned}\big|\mathbb{E}_A(P(X)) - \text{ABOI(N,M)}\big| &= \big|\mathbb{E}_A(P(X)) - P_M + P_M - \text{ABOI(N,M)}\big| \\ &\leq \underbrace{\big|\mathbb{E}_A(P(X)) - P_M\big|}_{(I)} + \underbrace{\big|P_M - \text{ABOI(N,M)}\big|}_{(II)} \end{aligned} \tag{5.12}$$

We will bound (I) and (II) separately.

$(I) : \big|\mathbb{E}_A(P(X)) - P_M\big|$

As stated in Equation (5.7), we can define $\mathbb{E}_A(P(X))$ as follows:

$$\mathbb{E}_A(P(X)) = \lim_{\Delta A \to 0} \frac{1}{m_{(\Delta A)}} \sum_{i=1}^{m_{(\Delta A)}} P(x_i) \tag{5.13}$$

or equivalently, for all $\varepsilon > 0$, there exists a $\delta > 0$ such that

$$\Delta A \leq \delta \implies \left| \mathbb{E}_A(P(X)) - \frac{1}{m_{(\Delta A)}} \sum_{i=1}^{m_{(\Delta A)}} P(x_i) \right| < \varepsilon/2 \tag{5.14}$$

where $\dfrac{1}{m_{(\Delta A)}} \displaystyle\sum_{i=1}^{m_{(\Delta A)}} P(x_i)$ is analogous to what we previously defined as P_M.

Therefore, hypothesis ($\star\star\star$) allows us to select a fine-grained enough grid M that gives us the desired error bound $\varepsilon/2$.

Selecting M as aforementioned, we have that

$$\left| \mathbb{E}_A(P(X)) - P_M \right| \leq \varepsilon/2 \tag{5.15}$$

$(II) : \left| P_M - \text{ABOI(N,M)} \right|$

As previously mentioned, P_M is the arithmetic mean of real signal strength over M (m unknown values), whereas ABOI(N,M) is the arithmetic mean of the m interpolated values on grid M obtained by OK interpolation of the original n measurements in N. Therefore, P_M and ABOI(N,M) are defined as follows:

$$P_M = \left(\sum_{i=1}^{m} P(x_i) \right) / m$$
$$\text{ABOI(N,M)} = \left(\sum_{i=1}^{m} I_N(x_i) \right) / m$$

where $P(\cdot)$ corresponds to the real signal strength, and $I_N(\cdot)$ corresponds to the OK interpolation of the original n signal strength measurements in N. Accordingly,

$$
\begin{aligned}
\big|P_M - \text{ABOI(N,M)}\big| &= \left| \left(\sum_{i=1}^{m} P(x_i) \right) /m - \left(\sum_{i=1}^{m} I_N(x_i) \right) /m \right| \\
&= \left| \left(\sum_{i=1}^{m} P(x_i) - I_N(x_i) \right) /m \right| \\
&\leq \left(\sum_{i=1}^{m} \big|P(x_i) - I_N(x_i)\big| \right) /m \\
&\leq \max_{i \in [1:m]} \big|P(x_i) - I_N(x_i)\big|
\end{aligned}
$$

Thus, the difference between P_M and ABOI(N,M) is bounded by the maximum interpolation error among all the m positions of grid M. Wang et al. [167] provided an exhaustive analysis regarding this maximum interpolation error of OK. Indeed, hypothesis ($\star$) allows the use of Corollary 1 of Wang et al. [167] along with Theorem 11.22 of Wendland et al. [170] to obtain the following result (a detailed description of this outcome is provided in Appendix B):

$$
\lim_{h_N \to 0} \mathbb{E}\Big[\max_{i \in [1:m]} \big|P(x_i) - I_N(x_i)\big| \Big] = 0 \tag{5.16}
$$

or equivalently, for all $\varepsilon > 0$, there exists a $\bar{h}$ such that

$$
h_N \leq \bar{h} \implies \mathbb{E}\Big[\max_{i \in [1:m]} \big|P(x_i) - I_N(x_i)\big| \Big] \leq \varepsilon/2
$$

Therefore, hypothesis ($\star\star$) gives us the conditions such that h_N is small enough to guarantee the desired error bound $\varepsilon/2$:

$$
\mathbb{E}\Big[\big|P_M - \text{ABOI(N,M)}\big| \Big] \leq \varepsilon/2 \tag{5.17}
$$

Finally, by joining the bounds for (I) and (II), i.e., by plugging (5.15) and (5.17) into (5.12), we obtain the desired inequality

$$
\mathbb{E}\Big[\big|\mathbb{E}_A(P(X)) - \text{ABOI(N,M)}\big| \Big] \leq \varepsilon
$$

which completes the proof.

□

5.5.2 Improvement on the Arithmetic Mean

In this section, we show that the ABOI method is an improvement on the arithmetic mean in estimating $\overline{P}_A$. Both methods require specific conditions about the number and position of the initial signal strength measurements. On the one hand, ABOI requires h_N to be small enough (hypothesis ($\star\star$) of Theorem 2). On the other hand, the arithmetic mean requires the initial measurements to fulfill Monte Carlo integration hypotheses. Only if satisfying these conditions, the methods can be considered as appropriate for estimating the mathematical expectation of signal strength. In the following, we will show that:

Preposition 3 If the initial measurements allow the arithmetic mean to be considered as an appropriate estimator of $\overline{P}_A$, this implies that the ABOI method will also be considered as an appropriate estimator of $\overline{P}_A$.

Preposition 4 If the initial measurements allow the ABOI method to be considered as an appropriate estimator of $\overline{P}_A$, this does not imply that the arithmetic mean will be considered as an appropriate estimator of $\overline{P}_A$.

Proof of Preposition 3. If the initial conditions allow the arithmetic mean to be considered as an appropriate estimator of $\overline{P}_A$, then the set of n signal strength measurements fulfills Monte Carlo integration hypotheses (Section 5.4.1). That is, the number n of measurements is high enough, and they are independent and uniformly distributed over the area. Theorem 6.6 of Niederreiter et al. [119] suggests a bound for h_N derived from its (extreme) discrepancy $D_n(N)$,

$$h_N \leq \sqrt{2} D_n^{1/2}(N)$$

where N is the set of positions of the n initial measurements. Given that the positions in N are independent random variables uniformly distributed over the area, Pronzato [132] states that

$$D_n(N) = \mathcal{O}[(\log n)^2/n]$$

This result indicates that, after a given number of measurements, hypothesis ($\star\star$) will be satisfied. Therefore, the ABOI method will also be considered as an appropriate estimator of $\overline{P}_A$.

□

The intuition behind Preposition 3 is that in case of measurements uniformly distributed over the area, both methods can be considered as appropriate to estimate $\overline{P}_A$. However, as discussed in Section 5.1, uniform spatial distribution is an unrealistically optimistic case for crowdsourced measurements.

Proof of Preposition 4. If the conditions allow the ABOI method to be considered as an appropriate estimator of $\overline{P}_A$, then hypothesis ($\star\star$) is fulfilled. This hypothesis only requires h_N to be small enough and does not require any specific distribution of measurements over the area. In particular, it does not require the measurements to be independent nor uniformly distributed over the area, which are necessary conditions for fulfilling Monte Carlo integration hypotheses. Therefore, the arithmetic mean may not be considered as an appropriate estimator of $\overline{P}_A$.

□

The intuition behind Preposition 4 is that the requirements of ABOI are less restrictive and more likely to be true in Mobile Crowdsourcing scenarios. As mentioned before, uniform spatial distribution is not a realistic case for measurements taken by real users, and therefore, there is no mathematical foundation for using the arithmetic mean (Section 5.4.1). However, real crowdsourced data is still able to fulfill hypothesis ($\star\star$), so far as the number of measurements allows it. Indeed, it is certainly expected that if the number of measurements is very low, then ABOI's estimation will not be accurate, since h_N will hardly be small enough. Likewise, if the number of signal strength measurements is high, then ABOI's estimation will be inclined to be closer to $\overline{P}_A$. This is a recurrent condition when estimating values of random effect models from measurements, and therefore, it can not be avoided due to the stochastic behavior of the observations.

5.6 Experimental Results

Given that Section 5.5 provided the mathematical foundations for using the ABOI method to estimate the expected signal strength value, we wanted to analyze its suitableness for this task experimentally. Additionally, we wanted to compare ABOI against the other aggregation methods commonly employed to estimate the mathematical expectation of signal strength (Section 5.4). In particular, we were interested in comparing ABOI with the arithmetic mean, as Section 5.5.2 gives us the intuition that the estimations provided by the ABOI method should be at least as good as the estimations provided by the arithmetic mean.

To evaluate and compare the aggregation methods described in Section 5.4, we performed experiments in both simulated and real scenarios. As this work is the first attempt to challenge existing assumptions about signal strength aggregation, we performed the following simplifications to the problem of estimating the mathematical expectation of signal strength in an area:

1. We considered areas with signal strength coming from only one base transceiver station (BTS).
2. Even when there may be a time variability of signal strength in the area [149, 93], we considered that the mathematical expectation is estimated for a static power configuration of the BTS.

5.6.1 Simulated Scenario

We considered an area A of 500 m $\times$ 500 m where a 30-meter tall BTS is placed at the center. We simulated the real signal strength on a fine-grained grid G over A with 5 m spacing, considering long-term attenuation due to path loss equation and medium-term variation due to shadowing modeled by a full covariance matrix [83, 129, 51, 144, 149]. Indeed, the real signal strength in G is given by

$$\vec{1}P - 10\alpha \log_{10}(\vec{d}) + \vec{v} \tag{5.18}$$

where $\vec{1}P = [P, P, ..., P]^T$ is a vector with n repeated values of P, the power transmitted by the simulated BTS; α corresponds to the path loss exponent; and $10\alpha \log_{10}(\vec{d})$ is the path loss attenuation, where $\vec{d} = [d_1, d_2, ..., d_n]^T$ is the vector of distances between the position of each measurement and the position of the BTS. In addition, $\vec{v}$ is an attenuation factor due to shadowing effects, where

$$\vec{v} \sim \mathcal{N}(0, \Sigma_v)$$

and the covariance matrix Σ_v is composed of elements given by $\text{Cov}\,(x_i, x_j) = \sigma_v^2\,(-d_{ij}/D_{corr})$, where d_{ij} is the distance between the positions x_i and x_j in G, and D_{corr} is a parameter that models the correlation among the measurements.

Next, we simulated signal strength measurements as if they were taken by real mobile devices. That is, the measurements included long-term attenuation due to path loss equation and medium-term variation due to shadowing, but they also included accuracy errors in both signal strength values and geographic coordinates (due to hardware inaccuracy). The simulated measurements are given by

$$\vec{X} = \vec{1}P - 10\alpha \log_{10}(\vec{d}) + \vec{u} + \vec{v} + \vec{w} \tag{5.19}$$

where $\vec{X} = [x_1, x_2, ..., x_n]^T$ is an $n \times 1$ vector that contains the measurements. As in Equation (5.18), $\vec{v}$ is the attenuation factor due to shadowing effects. Additionally, as geolocation sensors are not perfectly accurate, position errors are considered when estimating the position of each measurement. This component is simulated by

$$\vec{u} \sim \mathcal{N}(0, \rho_u^2 D),$$

which corresponds to a Gaussian distribution with a mean vector $\vec{0}$ and covariance matrix $\rho_u^2 D$, where $D = \text{diag}\{1/d_1, 1/d_2, ..., 1/d_n\}$ [149]. Finally, $\vec{w}$ in Equation (5.19) is some unrelated additive noise, where

$$\vec{w} \sim \mathcal{N}(0, \sigma_w^2 I_n).$$

For this simulation, the following values were used: $P = -10$ dBm, $\alpha = 3.5$, $\sigma_w = \sqrt{7}$ dB, $\sigma_v = \sqrt{10}$ dB, $\rho_u = 0.2$ dB and $D_{corr} = 50$ m. This setting is the same one used by Santos et al. [149].

Thus, signal strength values simulated over grid G using Equation (5.19) generate the spatial field shown in Figure 5.2.

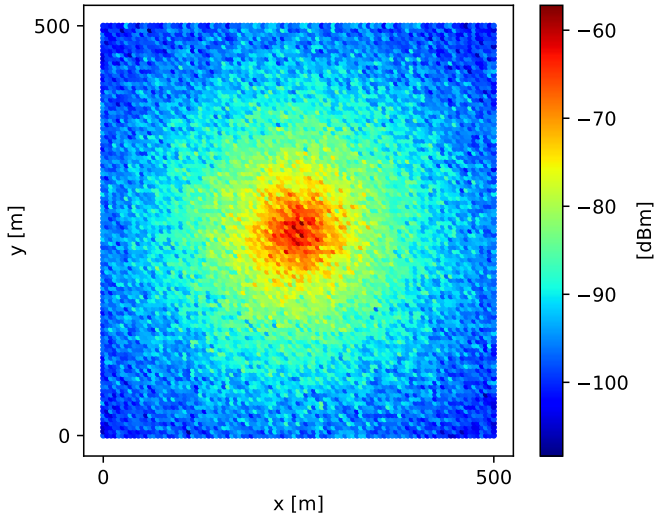


Figure 5.2: Simulated spatial field of signal strength over a fine-grained grid G.

We calculated the ground truth $\overline{P}_A$ (expected value of signal strength over A) as a Riemann sum considering all values in G.

It is important to clarify that, although the simulation model and its parameters were defined using dBm values, we always carefully manipulated signal strength values using the linear watt scale. Thus, we avoided the mishandling of log-scaled signal strength values, as discussed in Section 5.3.

For this experiment, we took different signal strength measurement sets of sizes 50, 100, 200, 400, 700, and 1000. We distributed the samples on the grid by using two different methods:

1. Completely uniform distribution on the grid, which is commonly used, but not realistic for Mobile Crowdsourcing scenarios, as discussed in Section 5.1.

2. Considering the mobility model based on social network theory proposed by Musolesi et al. [113]. This model is closer to the spatial distribution of Mobile Crowdsourcing measurements, as they are defined by human mobility.

Figure 5.3 shows the difference in the spatial distribution of 100 samples using the two methods explained above.

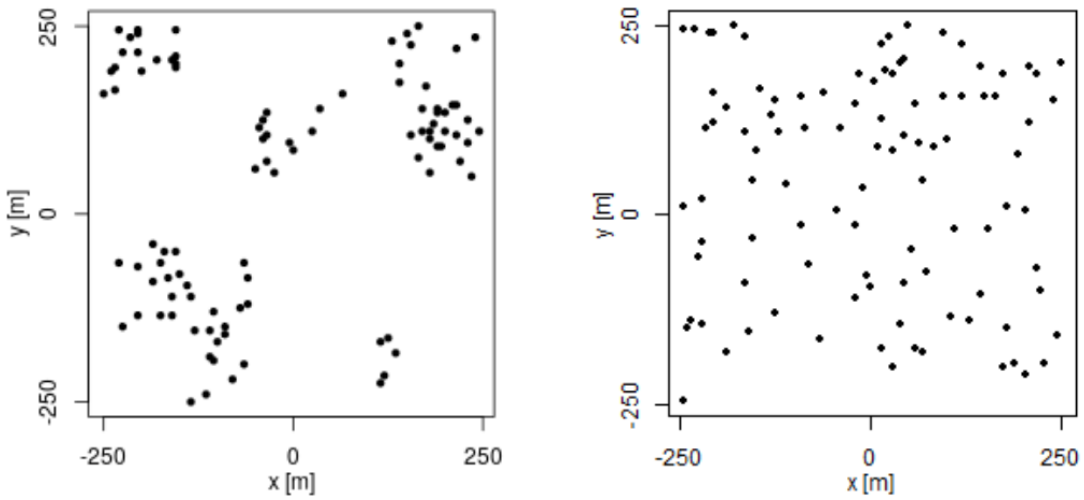
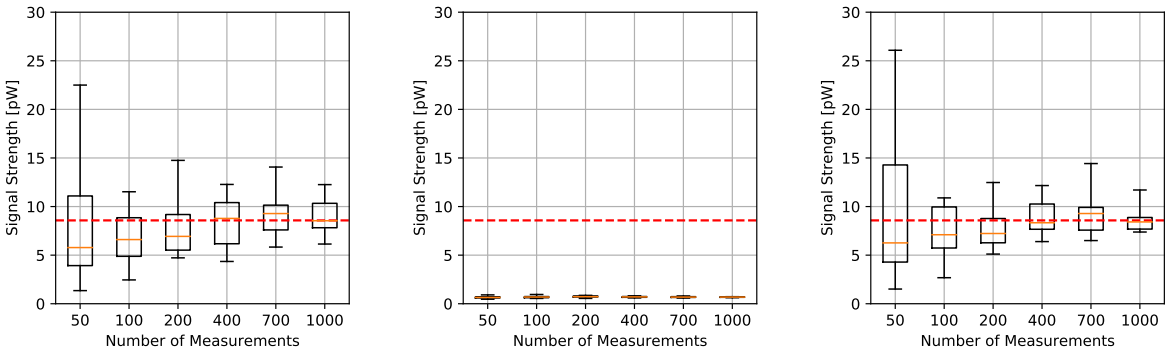


Figure 5.3: Example of spatial distribution for 100 samples using uniform distribution (left) and distribution based on social network theory (right).

For every sample size and type of spatial distribution, we estimated $\overline{P}_A$ by applying the three methods presented in Section 5.4. We repeated each experiment 40 times, i.e., we took 40 different sample sets in every case.

The results for experiments using uniform distribution are shown in Figure 5.4. For each aggregation method and sample size, we have the boxplot that depicts the estimations of $\overline{P}_A$. It is important to clarify that all figures were calculated in linear scale, avoiding the errors mentioned in Section 5.3. All signal strength values are shown in pW units, where 1 pW = 1×10^{-12} W.



(a) Arithmetic Mean (b) Median Value (c) ABOI Method

Figure 5.4: Simulated scenario. Boxplots for $\overline{P}_A$ prediction using the three aggregation methods and different sample sizes, selected by uniform distribution. Real $\overline{P}_A$ value in the red line.

As expected, arithmetic mean estimations tended to be close to $\overline{P}_A$, since uniform distribution is its best case, as explained in Section 5.4.1. The median value performed poorly, predicting nearly constant values far from the real one. Our proposed ABOI method showed satisfactory results and a similar behavior to the arithmetic mean.

In addition, Figure 5.5 shows the RMSE measure obtained by the aggregation methods, properly calculated using the linear values of the estimations of $\overline{P}_A$, as discussed in Section 5.3.2. RMSE values for our proposed ABOI rapidly decreased to low values, obtaining very similar results to the arithmetic mean. These results agree with the intuition of Preposition 3, as in the case of uniform spatial distribution, both ABOI and the arithmetic mean performed well in estimating $\overline{P}_A$.

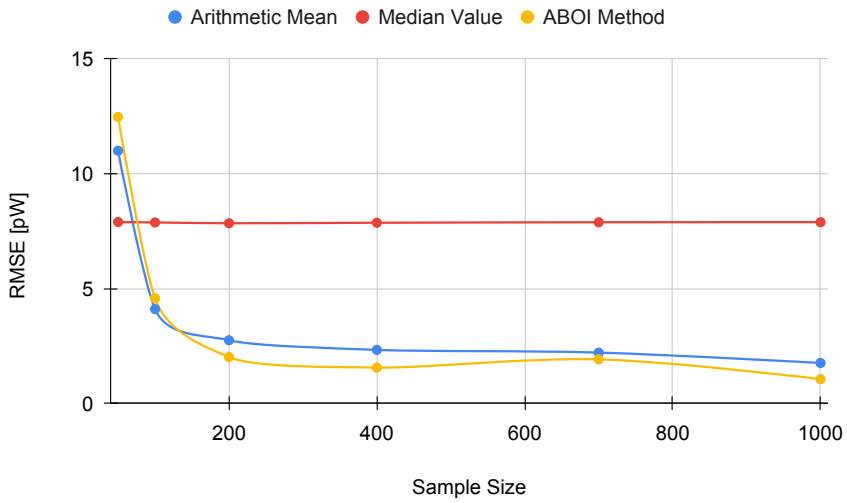
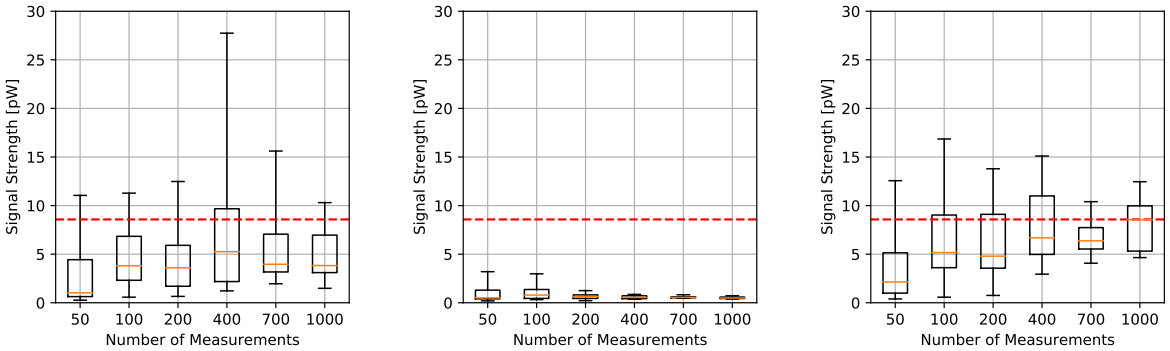


Figure 5.5: Simulated scenario. RMSE for $\overline{P}_A$ prediction for different sample sizes with uniform distribution.

The results for experiments using spatial distribution based on social network theory are shown in Figure 5.6. The arithmetic mean was more erratic than before, without a clear convergence to real $\overline{P}_A$ as the sample size increased. The median value showed similar behavior to the uniform distribution case, predicting nearly constant values. The ABOI method showed again a tendency to be close to real $\overline{P}_A$, but with a higher variability than for uniform distribution.



(a) Arithmetic Mean (b) Median Value (c) ABOI Method

Figure 5.6: Simulated scenario. Boxplots for $\overline{P}_A$ prediction using the three aggregation methods and different sample sizes, selected by distribution based on social network theory. Real $\overline{P}_A$ value in the red line.

Figure 5.7 shows that our proposed ABOI method obtained consistently lower RMSE values than the other methods, with a remarkable improvement over the arithmetic mean. Therefore, these experiments in a simulated scenario showed that the ABOI method is more reliable and more independent of the spatial distribution of samples at estimating the mathematical expectation of signal strength.

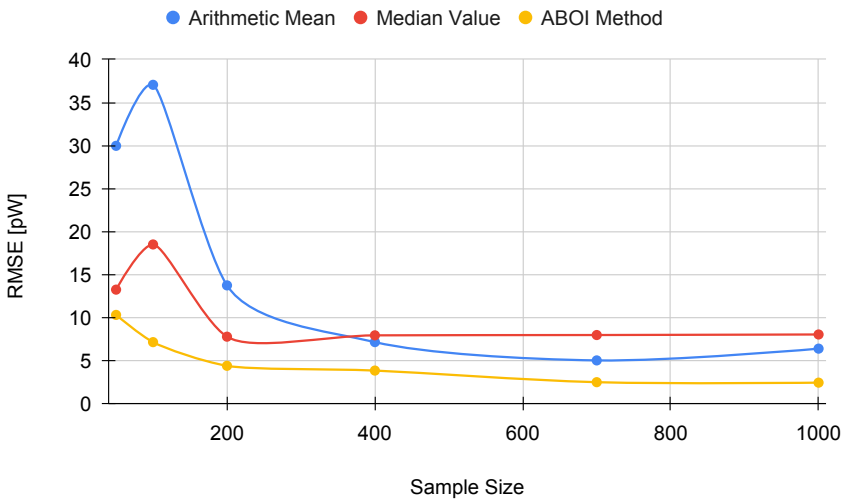


Figure 5.7: Simulated scenario. RMSE for $\overline{P}_A$ prediction for different sample sizes with distribution based on social network theory.

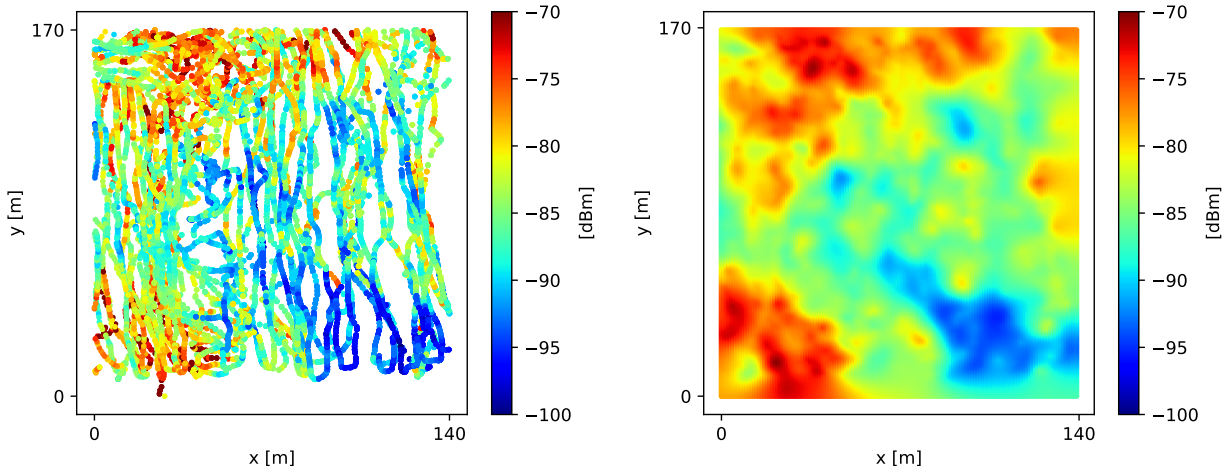
These results are also consistent with the mathematical foundations presented in Section 5.5, as ABOI performed well in estimating $\overline{P}_A$ in a nonuniform distribution scenario, which was close to the spatial distribution of crowdsourced measurements. In addition, as expected due to Proposition 4, spatial distribution based on social network theory did not satisfy the conditions required by the arithmetic mean to properly estimate $\overline{P}_A$.

5.6.2 Real Data

To test the aggregation methods using real data, we developed a very minimalist Android application to take signal strength measurements with a fine-grained time interval. The application was designed to run every 0.5 seconds. During each execution, the application used Android's Telephony Manager [7] to access information about the current cell being used by the device for network signaling. Thus, the Telephony Manager provided a `CellIdentity` object to obtain cell identifiers and a `CellSignalStrength` object to obtain the technology-specific signal strength in dBm. Along with this cell-related information, the application also stored the current location (latitude and longitude) with the highest accuracy possible.

During a period of 2 consecutive hours, we took near to 24,000 signal strength measurements around the vicinity of a single LTE BTS ($eNodeB$) located in a residential area using two different mobile devices. The received signal strength measurements densely covered an area of 140 m $\times$ 170 m near the BTS, as shown in Figure 5.8a. To calculate the $\overline{P}_A$ value,

we aligned the real measurements into a fine-grained grid G with 1 m spacing, obtaining the spatial field shown in Figure 5.8b. Then, we calculated the ground truth $\overline{P}_A$ as a Riemann sum of all values in G.



(a) Position of real measurements

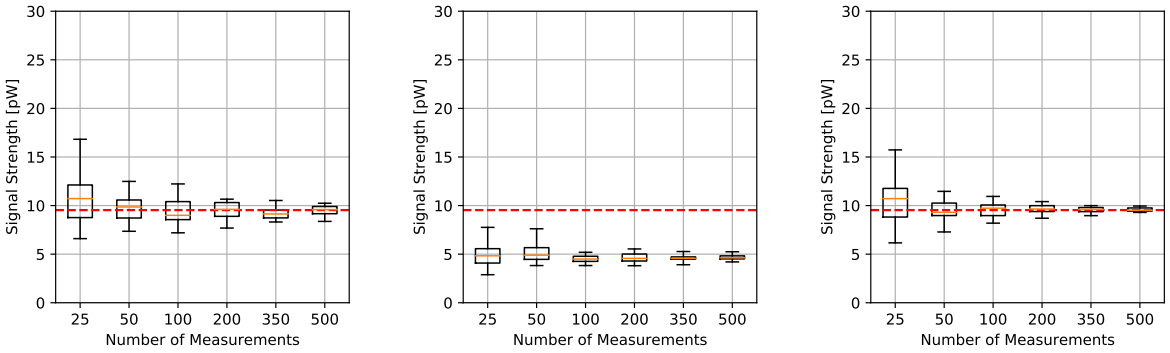
(b) Interpolated spatial field

Figure 5.8: Real signal strength around the vicinity of a single LTE BTS. Color represents the dBm value of samples.

Similarly to the simulation case, we performed experiments for different sample sets of sizes 25, 50, 100, 200, 350, and 500. We also considered both spatial distribution methodologies: uniform distribution and based on social network theory. As for the simulation experiments, we repeated each experiment 40 times.

The results for experiments using uniform distribution are shown in Figure 5.9. We found that the behavior of the three methods was similar to the behavior shown by themselves in the simulation case with uniform distribution (Figure 5.4). The arithmetic mean and the ABOI method presented low and similar variability and fast convergence to the calculated value of $\overline{P}_A$, where the ABOI method obtained slightly closer estimations to $\overline{P}_A$. The median value also showed coincident behavior with the simulation case, predicting nearly constant and low values far from $\overline{P}_A$.

Figure 5.10 confirms our analysis, as both the arithmetic mean and our proposed ABOI method obtained similar RMSE values, outperforming the median value. It is important to remember that, as stated in Section 5.4.1, uniform spatial distribution is the best case for the arithmetic mean, and therefore, its good performance was expected.



(a) Arithmetic Mean (b) Median Value (c) ABOI Method

Figure 5.9: Real scenario. Boxplots for $\overline{P}_A$ prediction using the three aggregation methods and different sample sizes, selected by uniform distribution. Calculated $\overline{P}_A$ value in the red line.

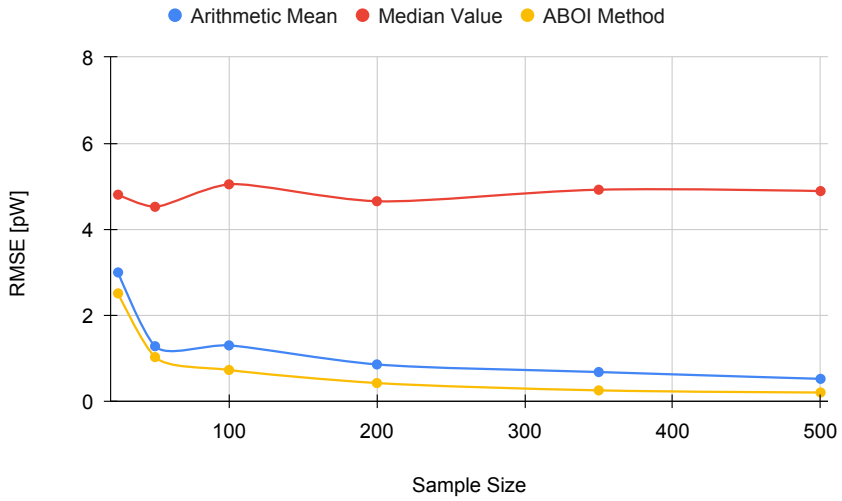
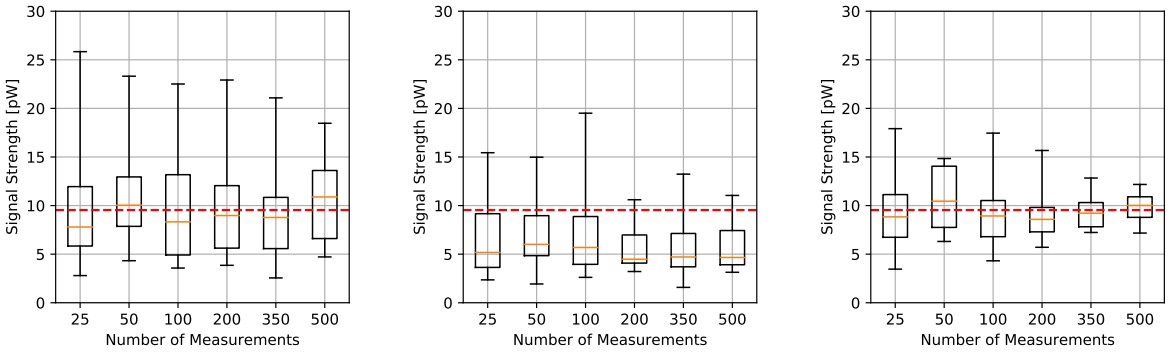


Figure 5.10: Real scenario. RMSE for $\overline{P}_A$ prediction for different sample sizes with uniform distribution.

As for the simulated scenario, these results are coherent with the intuition of Preposition 3, as in the case of uniform spatial distribution, both ABOI and the arithmetic mean performed well in estimating $\overline{P}_A$.

The results for experiments with spatial distribution based on social network theory are shown in Figure 5.11. The arithmetic mean showed higher variability and worse estimations of $\overline{P}_A$ in relation to the previous case. The median value tended to predict low values. Our proposed ABOI method showed a similar behavior to the uniform distribution case, showing a clear convergence to $\overline{P}_A$. It also presented lower variability than the arithmetic mean.

Figure 5.12 shows that the ABOI method obtained consistently lower RMSE values than the other methods, with a clear improvement over the arithmetic mean. Unlike the arith-



(a) Arithmetic Mean (b) Median Value (c) ABOI Method

Figure 5.11: Real scenario. Boxplots for $\overline{P}_A$ prediction using the three aggregation methods and different sample sizes, selected by distribution based on social network theory. Calculated $\overline{P}_A$ value in the red line.

metic mean, our proposed method obtained more stable RMSE values using both spatial distribution scenarios.

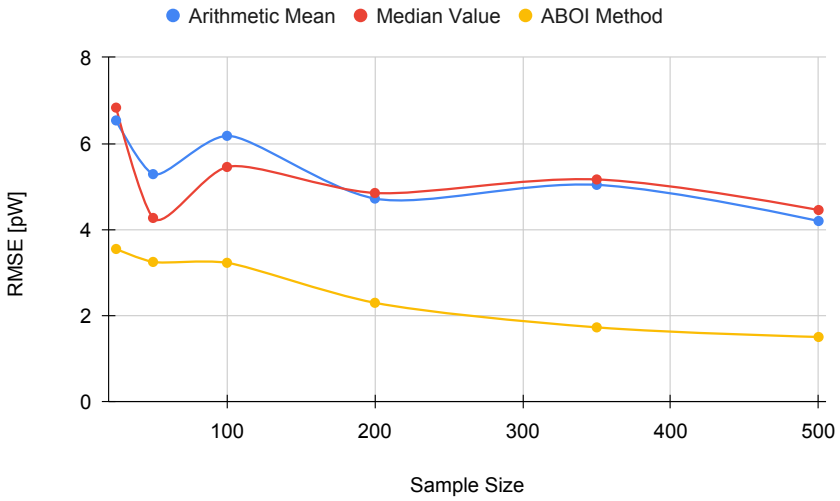


Figure 5.12: Real scenario. RMSE for $\overline{P}_A$ prediction for different sample sizes with distribution based on social network theory.

These results also agree with the mathematical foundations presented in Section 5.5 and with the obtained results in the simulated scenario. That is, the spatial distribution based on social network theory allowed ABOI to perform well in estimating $\overline{P}_A$. In contrast, this spatial distribution did not satisfy the conditions required by the arithmetic mean to properly estimate $\overline{P}_A$.

Therefore, these experiments in a real scenario showed that our proposed method is more reliable and more independent of the spatial distribution of samples at estimating the mathematical expectation of signal strength.

Regarding the algorithms' runtime performance, we measured their execution time using a 3.4GHz quad-core processor (Intel Core i5-3570) with 12GB RAM. For all methods, the execution time increased along with the number of measurements. On the one hand, the execution of the median value and the arithmetic mean never exceeded 0.01 seconds (both in the simulated case and in the real case). On the other hand, the execution of the ABOI method never exceeded 2 seconds. These results exemplify a well-known trade-off between estimation goodness and complexity. However, in our particular case, reducing the error in estimating $\overline{P}_A$ is much more relevant than reducing the time needed to compute the estimation, considering that the ABOI method's runtime is still very low. Therefore, we do not consider the execution time as a major drawback of our method.

5.7 Summary

In this chapter, we first presented the physical and mathematical formalities of how signal strength values must be handled at applying mathematical operations in a scientific and academic environment to avoid some common sources of error. We formally showed why some simple tasks, such as averaging and comparing signal strength values, are usually performed in contradiction with some scientific principles due to the indiscriminate use of log-scaled values. Indeed, these situations commonly lead to errors in the analysis of experimental data, and therefore, to make wrong conclusions.

In addition, we presented a novel method based on interpolation to aggregate signal strength samples into one representative value to estimate the mathematical expectation of signal strength in an area. This method is shown to present solid mathematical foundations to be employed in real Mobile Crowdsourcing scenarios.

Our proposed ABOI method outperformed other commonly used aggregation methods, as the arithmetic mean and the median value, mainly because it was shown to be more independent of some Mobile Crowdsourcing data difficulties such as nonuniform spatial distribution of the samples, the potentially low number of measurements and the inaccuracy of end-user devices. By using this method, we computed more reliable estimations of the mathematical expectation of signal strength in both simulated and real scenarios.

We conclude that for most Mobile Crowdsourcing scenarios, our proposed ABOI method should be preferred over the other methodologies. Nevertheless, the ABOI method could be improved by considering more complex simulated scenarios, such as areas with multiple antennas, and taking into account small-scale fading caused by multipath propagation, and short-term attenuation fluctuations due to time variance in the channel. Additionally, as mentioned in Section 5.4.3, our proposed method could be improved using a more complex and accurate interpolation algorithm.

Chapter 6

Conclusion

Predicting the network state is of great interest to the networking community, especially when it can help maintain the system's reliability or enhance user experience. Moreover, the proposal of anticipatory networking methodologies is significant, as it could positively impact network optimization at different layers of the networking system.

In this thesis work, we addressed two crucial anticipatory networking problems. Firstly, we focused on detecting DNS anomalies inside critical DNS servers. This problem has great relevance, since one of the main requirements of the DNS is its availability, and therefore, there is increasing concern over its vulnerability to attacks and failures. Secondly, we focused on the prediction of throughput for mobile network users. Given the emergence of new challenges and critical requirements related to the evolution of mobile networks, there has been an increasing concern about user throughput prediction, since accurate throughput predictions are challenging but essential for the efficiency of critical networking applications.

6.1 Achievement of Objectives

During the development of this thesis, we worked towards the validation of the research hypothesis: *It is possible to improve the accuracy of short-term prediction in real anticipatory networking problems, by using online time series analysis and its adaption against anomalies and concept drifts.* The results presented in this work confirmed the proposed hypothesis, i.e., the hypothesis is supported by the contributions described throughout the chapters of this thesis regarding the two selected anticipatory networking problems.

In the case of the anticipatory networking problem related to DNS traffic, we did improve the detection of anomalies in DNS traffic with respect to state-of-the-art baseline methodologies. The proposed method is based on the prediction of DNS traffic statistics, using an online multivariate time series forecasting approach. The prediction model is designed to be continuously updated, which is essential to enhance the prediction of DNS traffic statistics against anomalies and concept drifts in the traffic data. Our anomaly detection methodology exhibits outstanding performance in different real-world scenarios, and therefore, it can be

helpful for DNS operators to improve the reliability of their services.

Regarding the anticipatory networking problem related to mobile QoS, we did improve throughput prediction for mobile networks with respect to state-of-the-art baseline methodologies. We based our prediction approaches on a deep inspection of the empirical effect of the SINR on user throughput by proposing an SINR-dependent probabilistic mixture model to estimate the probability distribution of user throughput for any particular SINR level. Then, we proposed two prediction strategies, which directly estimate the mathematical expectation of the probability distribution of user throughput. These strategies are useful for forecasting the user throughput time series in real time, covering a wide range of real-world application scenarios. The proposed prediction methodologies can be helpful to face anomalies and concept drifts phenomenon, such as the presence of handovers, in which the user equipment changes its connectivity to a different mobile network technology.

In the following, for each of the two anticipatory networking problems, we review in detail the achievement of the research objectives A, B, C, and D, previously described in Section 1.2.

6.1.1 Anticipatory Networking for DNS

A. **To obtain a proper representation of the data to be analyzed and predicted.**
In Section 2.3, we proposed aggregating DNS traffic data into different groups. For each DNS traffic window, we computed a set of nine features, allowing us to represent the DNS traffic as a nine-dimensional multivariate time series, which was then analyzed and predicted. Moreover, in Section 2.6, this aggregation process was shown to be essential for detecting a wide range of DNS anomalies.

B. **To establish accuracy measures to be used for comparison between different prediction models.**
In Section 2.4, we discussed the limitation of testing DNS server monitoring and troubleshooting tools because of the lack of labeled anomalies. Therefore, we designed a tool to generate a minimal set of data points that a DNS anomaly detector would be expected to detect. Then, in Section 2.6, we evaluate the different anomaly detection models according to their sensitivity configurations needed to detect the injected anomalies.

C. **To clearly identify the limits of predictability for each problem.**
In Section 2.7.5, we discussed the lack of labeled data for DNS anomaly detection and its implications. Indeed, this issue could result in training our forecast model using DNS traffic with some anomalies inside. We can then implicitly make the model learn these anomalies and use them to define normal traffic behavior. Moreover, we discussed the challenge of having recurrent traffic anomalies which appear to become part of normal traffic.

D. **To develop a predictive model that considers network anomalies to adapt its predictions.**
Along Chapter 2, we present and validate a near real-time Anomaly Detection Based on Prediction (AD-BoP) method, providing a useful and easily explainable methodology to detect DNS anomalies effectively.

6.1.2 Anticipatory Networking for mobile QoS

A. **To obtain a proper representation of the data to be analyzed and predicted.**
In Chapter 3, we performed a preliminary study covering a broad range of mobile network measurements, which led to a more detailed examination of the relationship between user throughput and signal quality. Then, in Chapter 4, we thoroughly analyze the impact of channel quality variations on user throughput using a novel approach. Indeed, we study the user throughput as a random variable that depends on the current signal-to-interference-plus-noise ratio (SINR). Thus, we model the distribution of user throughput as an SINR-dependent probabilistic mixture model that properly fits the empirical data.

B. **To establish accuracy measures to be used for comparison between different prediction models.**
Guided by the related literature, in Section 4.9, we compared the different throughput prediction approaches by computing the root-mean-squared error (RMSE) and mean absolute error (MAE). In order to extend our analysis, we also considered different time scales for the user throughput time series. Thus, we compared the prediction methods by covering a wide range of target application requirements.

C. **To clearly identify the limits of predictability for each problem.**
In Section 3.6, we discussed several sources of variability that limit the predictability of mobile QoS. Indeed, we analyzed the effect of some internal and external variables that impact the performance of network connections. Then, as some of these variables can not be controlled, it is crucial to consider these sources of variability when deploying mobile QoS prediction models.

D. **To develop a predictive model that considers network anomalies to adapt its predictions.**
In Section 4.8, we proposed two prediction strategies, which directly estimate the mathematical expectation of the probability distribution of user throughput. The prediction models are mainly based on an SINR-dependent probabilistic mixture model. The adoption of our proposed methodologies is beneficial for practical applications since they cover a wide range of application scenarios, are easily explainable, and can be deployed on regular mobile devices as they only rely on the observed SINR.

6.2 Future Work

In this thesis work, we focused on providing solutions to two key anticipatory networking problems, which are essential for preserving the well-working of all Internet-based resources and improving user experience. The proposed method for detecting DNS anomalies is based on the prediction of DNS traffic, and therefore, a different time series forecasting strategy could be used to improve its detection accuracy. Additionally, more experiments could be carried out to find optimal values for the method's configuration, such as the sensitivity threshold and the length of traffic windows. In the case of our proposed throughput prediction methodology, it is based on estimating the expected value of user throughput for a given channel quality level. However, this approach could be used for implementing a

fully stochastic throughput prediction strategy, where the throughput estimation (expected value) is accompanied by a confidence level computed directly from the estimated probability density function of user throughput. Then, the target application (e.g., adaptive bitrate streaming) could use that information to make a probabilistic decision about the predicted throughput value, taking into account the different drawbacks caused by under-estimation and over-estimation of the user throughput.

The solutions provided for these problems are fundamental as they address two crucial issues related to network quality. Moreover, these solutions encompass a broad range of networking scenarios, and therefore, they serve as a basis for addressing a variety of other anticipatory networking problems. Firstly, our proposed solutions are based on machine learning and statistical modeling for time series forecasting, which is essential, as most network data can be naturally represented as a time series. Secondly, we considered two networking problems related to different network layers in the seven-layer OSI model of computer networking, dealing with network information close to the user and network information close to the physical channel.

Therefore, the analysis presented in this thesis can be extrapolated to other related anticipatory networking problems. In particular, the solution provided to the DNS anomaly detection problem could serve as a basis for detecting anomalies in network traffic at any capture point on the network, including users' equipment, content servers, and network probes. This procedure could be beneficial to perform early detection of critical threats at different network levels. Regarding the solution provided to the throughput prediction problem, it can be helpful for predicting other metrics related to network performance, such as jitter and latency. Moreover, this prediction approach can be employed to forecast throughput at different capture points on the network, in order to improve the efficiency of different networking applications, such as the resource scheduling process in mobile base stations.

Bibliography

[1] Mohiuddin Ahmed and Abdun Naser Mahmood. Novel approach for network traffic pattern analysis using clustering-based collective anomaly detection. *Annals of Data Science*, 2(1):111–130, 2015.

[2] Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, 2016.

[3] Emmanouil Alimpertis, Athina Markopoulou, Carter Butts, and Konstantinos Psounis. City-wide signal strength maps: Prediction with random forests. In *The World Wide Web Conference*, pages 2536–2542. ACM, 2019.

[4] Roberto Alonso, Raúl Monroy, and Luis A Trejo. Mining ip to domain name interactions to detect dns flood attacks on recursive dns servers. *Sensors*, 16(8):1311, 2016.

[5] Marios Anagnostopoulos, Georgios Kambourakis, Stefanos Gritzalis, and David KY Yau. Never say never: Authoritative tld nameserver-powered dns amplification. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9. IEEE, 2018.

[6] Android. Sdk tools: dumpsys. `https://developer.android.com/studio/command-line/dumpsys`. [Online; accessed 18-April-2023].

[7] Android. Telephony manager. `https://developer.android.com/reference/android/telephony/TelephonyManager`. [Online; accessed 18-April-2023].

[8] Android. Vpnservice. `https://developer.android.com/reference/android/net/VpnService`. [Online; accessed 18-April-2023].

[9] Android. Workmanager. `https://developer.android.com/topic/libraries/architecture/workmanager`. [Online; accessed 18-April-2023].

[10] Somaya Arianfar. Tcp's congestion control implementation in linux kernel. In *Proceedings of Seminar on Network Protocols in Operating Systems*, page 16, 2012.

[11] Tahmina Azmin, Mohamad Ahmadinejad, and Nashid Shahriar. Bandwidth prediction in 5g mobile networks using informer. In *2022 13th International Conference on Network of the Future (NoF)*, pages 1–9. IEEE, 2022.

[12] Paramvir Bahl and Venkata N Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064)*, volume 2, pages 775–784. Ieee, 2000.

[13] Ali C Begen, Mehmet N Akcay, Abdelhak Bentaleb, and Alex Giladi. Adaptive streaming of content-aware-encoded videos in dash. js. *SMPTE Motion Imaging Journal*, 131(4):30–38, 2022.

[14] Peter Benko, Gabor Malicsko, and Andras Veres. A large-scale, passive analysis of end-to-end tcp performance over gprs. In *IEEE INFOCOM 2004*, volume 3, pages 1882–1892. IEEE, 2004.

[15] Anastasiia Beznosyk, Peter Quax, Karin Coninx, and Wim Lamotte. Influence of network delay and jitter on cooperation in multiplayer games. In *Proceedings of the 10th international conference on virtual reality continuum and its applications in industry*, pages 351–354, 2011.

[16] Arkadiusz Biernacki. Improving streaming video with deep learning-based network throughput prediction. *Applied Sciences*, 12(20):10274, 2022.

[17] Jean-Yves Bisiaux. Dns threats and mitigation strategies. *Network Security*, 2014(7):5–9, 2014.

[18] Mate Boban, Chunxu Jiao, and Mohamed Gharba. Measurement-based evaluation of uplink throughput prediction. In *2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring)*, pages 1–6. IEEE, 2022.

[19] S. Bortzmeyer. Dns privacy considerations. RFC 7626, RFC Editor, August 2015.

[20] Robert Braden. Requirements for internet hosts - communication layers. STD 3, RFC Editor, October 1989. `http://www.rfc-editor.org/rfc/rfc1122.txt`.

[21] Percy Williams Bridgman. *Dimensional formulas*, chapter 2. Yale university press, 1922.

[22] Percy Williams Bridgman. *The use of dimensional formulas in changing units*, chapter 3. Yale university press, 1922.

[23] Edy Budiman and Oki Wicaksono. Measuring quality of service for mobile internet services. In *2016 2nd International Conference on Science in Information Technology (ICSITech)*, pages 300–305. IEEE, 2016.

[24] Nicola Bui, Foivos Michelinakis, and Joerg Widmer. A model for throughput prediction for mobile users. In *European Wireless 2014; 20th European Wireless Conference*, pages 1–6. VDE, 2014.

[25] Nicola Bui and Joerg Widmer. Data-driven evaluation of anticipatory networking in lte networks. *IEEE Transactions on Mobile Computing*, 17(10):2252–2265, 2018.

[26] Pedro Casas, Alessandro D'Alconzo, Pierdomenico Fiadino, Arian Bär, Alessandro Finamore, and Tanja Zseby. When youtube does not work—analysis of qoe-relevant degradation in google cdn traffic. *IEEE Transactions on Network and Service Management*, 11(4):441–457, 2014.

[27] Pedro Casas, Andreas Sackl, Raimund Schatz, Lucjan Janowski, John Turk, and Ralf Irmer. On the quest for new kpis in mobile networks: The impact of throughput fluctuations on qoe. In *2015 IEEE International Conference on Communication Workshop (ICCW)*, pages 1705–1710. IEEE, 2015.

[28] Pedro Casas, Michael Seufert, Florian Wamser, Bruno Gardlo, Andreas Sackl, and Raimund Schatz. Next to you: Monitoring quality of experience in cellular networks from the end-devices. *IEEE Transactions on Network and Service Management*, 13(2):181–196, 2016.

[29] Milan Čermák, Pavel Čeleda, and Jan Vykopal. Detection of dns traffic anomalies in large networks. In *Meeting of the European Network of Universities and Companies in Information and Communication Engineering*, pages 215–226. Springer, 2014.

[30] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[31] Chris Chatfield. The holt-winters forecasting procedure. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 27(3):264–279, 1978.

[32] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.

[33] Chih-Chuan Cheng and Pi-Cheng Hsiu. Extend your journey: considering signal strength and fluctuation in location-based applications. *IEEE/ACM Transactions on Networking*, 23(2):451–464, 2014.

[34] Jerome T Connor, R Douglas Martin, and Les E Atlas. Recurrent neural networks and robust time series prediction. *IEEE transactions on neural networks*, 5(2):240–254, 1994.

[35] Luca Deri, Lorenzo Luconi Trombacchi, Maurizio Martinelli, and Daniele Vannozzi. A distributed dns traffic monitoring system. In *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 30–35. IEEE, 2012.

[36] Habiba Elsherbiny, Hazem M Abbas, Hatem Abou-zeid, Hossam S Hassanein, and Aboelmagd Noureldin. 4g lte network throughput modelling and prediction. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pages 1–6. IEEE, 2020.

[37] Ericsson. Ericsson mobility report. `https://www.ericsson.com/4ae28d/assets/local/reports-papers/mobility-report/documents/2022/ericsson-mobility-report-november-2022.pdf`, 2022. [Online; accessed 18-April-2023].

[38] Engin Eyceyurt, Yunus Egi, and Josko Zec. Machine-learning-based uplink throughput prediction from physical layer measurements. *Electronics*, 11(8):1227, 2022.

[39] Adriano Faggiani, Enrico Gregori, Luciano Lenzini, Valerio Luconi, and Alessio Vecchio. Smartphone-based crowdsourcing for network monitoring: opportunities, challenges, and a case study. *IEEE Communications Magazine*, 52(1):106–113, 2014.

[40] Mah-Rukh Fida, Andra Lutu, Mahesh K Marina, and Özgü Alay. Zipweave: Towards efficient and reliable measurement based mobile coverage maps. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2017.

[41] Alexander Frömmgen, Jens Heuschkel, Patrick Jahnke, Fabio Cuozzo, Immanuel Schweizer, Patrick Eugster, Max Mühlhäuser, and Alejandro Buchmann. Crowdsourcing measurements of mobile network performance and mobility during a large scale event. In *International Conference on Passive and Active Network Measurement*, pages 70–82. Springer, 2016.

[42] Zhenghua Fu, Xiaoqiao Meng, and Songwu Lu. How bad tcp can perform in mobile ad hoc networks. In *Proceedings ISCC 2002 Seventh International Symposium on Computers and Communications*, pages 298–303. IEEE, 2002.

[43] Ana Galindo-Serrano, Berna Sayrac, Sana Ben Jemaa, Janne Riihijärvi, and Petri Mähönen. Automated coverage hole detection for cellular networks using radio environment maps. In *2013 11th International Symposium and Workshops on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, pages 35–40. IEEE, 2013.

[44] Ali A Ghorbani, Wei Lu, and Mahbod Tavallaee. *Network intrusion detection and prevention: concepts and techniques*, volume 47. Springer Science & Business Media, 2009.

[45] Prasanta Gogoi, DK Bhattacharyya, Bhogeswar Borah, and Jugal K Kalita. A survey of outlier detection methods in network anomaly identification. *The Computer Journal*, 54(4):570–588, 2011.

[46] Prasanta Gogoi, Bhogeswar Borah, and Dhruba K Bhattacharyya. Anomaly detection analysis of intrusion data using supervised & unsupervised approach. *Journal of Convergence Information Technology*, 5(1):95–110, 2010.

[47] Harm Griffioen and Christian Doerr. Taxonomy and adversarial strategies of random subdomain attacks. In *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. IEEE, 2019.

[48] Martin Grill, Tomáš Pevnỳ, and Martin Rehak. Reducing false positives of network anomaly detection by local adaptive multivariate smoothing. *Journal of Computer and System Sciences*, 83(1):43–57, 2017.

[49] Andrei Gurtov and Jouni Korhonen. Effect of vertical handovers on performance of tcp-friendly rate control. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8(3):73–87, 2004.

[50] Andrei Gurtov and Reiner Ludwig. Responding to spurious timeouts in tcp. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, volume 3, pages 2312–2322. IEEE, 2003.

[51] Brian Ferris Dirk Hähnel and Dieter Fox. Gaussian processes for signal strength-based location estimation. In *Proceeding of robotics: science and systems*, 2006.

[52] Zhifeng Han, Jianxin Liao, Qi Qi, Haifeng Sun, and Jingyu Wang. Radio environment map construction by kriging algorithm based on mobile crowd sensing. *Wireless Communications and Mobile Computing*, 2019, 2019.

[53] Shagufta Henna. A throughput analysis of tcp variants in mobile wireless networks. In *2009 Third International Conference on Next Generation Mobile Applications, Services and Technologies*, pages 279–284. IEEE, 2009.

[54] Gary T Henry. *Practical sampling*, volume 21. Sage, 1990.

[55] Robert L Hershey. Analysis of the difference between log mean and mean log averaging. *The Journal of the Acoustical Society of America*, 51(4A):1194–1197, 1972.

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

[57] Paul E. Hoffman and Patrick McManus. DNS Queries over HTTPS (DoH). RFC 8484, October 2018.

[58] Zi Hu, Liang Zhu, John Heidemann, Allison Mankin, Duane Wessels, and Paul E. Hoffman. Specification for DNS over Transport Layer Security (TLS). RFC 7858, May 2016.

[59] Junxian Huang, Qiang Xu, Birjodh Tiwana, Z Morley Mao, Ming Zhang, and Paramvir Bahl. Anatomizing application performance differences on smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 165–178. ACM, 2010.

[60] Christian Huitema, Sara Dickinson, and Allison Mankin. DNS over Dedicated QUIC Connections. RFC 9250, May 2022.

[61] MF Ibrahim and JD Parsons. Signal strength prediction in built-up areas. part 1: Median signal strength. *IEE Proceedings F (Communications, Radar and Signal Processing)*, 130(5):377–384, 1983.

[62] Agbotiname Lucky Imoize, Samuel Oluwatobi Tofade, Glory Uzuazobona Ughegbe, Francis Ifeanyi Anyasi, and Joseph Isabona. Updating analysis of key performance indicators of 4g lte network with the prediction of missing values of critical network parameters based on experimental data from a dense urban environment. *Data in Brief*, 42:108240, 2022.

[63] Jana Iyengar and Martin Thomson. QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000, May 2021.

[64] Van Jacobson. Congestion avoidance and control. *ACM SIGCOMM computer communication review*, 18(4):314–329, 1988.

[65] Van Jacobson, Bob Braden, and Dave Borman. Tcp extensions for high performance. RFC 1323, RFC Editor, May 1992. `http://www.rfc-editor.org/rfc/rfc1323.txt`.

[66] Ramakrishna Janaswamy. Median signal predictions. In *Radiowave propagation and smart antennas for wireless communications*, chapter 3, pages 67–97. Springer, Boston, MA, 2002.

[67] Hao Jiang and Constantinos Dovrolis. Passive estimation of tcp round-trip times. *ACM SIGCOMM Computer Communication Review*, 32(3):75–88, 2002.

[68] Xiaolan Jiang, Yi-Han Chiang, Yang Zhao, and Yusheng Ji. Plato: Learning-based adaptive streaming of 360-degree videos. In *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*, pages 393–400. IEEE, 2018.

[69] Matt Joras and Yang Chi. How facebook is bringing quic to billions. `https://engineering.fb.com/2020/10/21/networking-traffic/how-facebook-is-bringing-quic-to-billions/`. [Online; accessed 18-April-2023].

[70] Mansour J Karam and Fouad A Tobagi. Analysis of delay and delay jitter of voice traffic in the internet. *Computer Networks*, 40(6):711–726, 2002.

[71] Mohamed Kadhem Karray. User's mobility effect on the performance of wireless cellular networks serving elastic traffic. *Wireless Networks*, 17(1):247–262, 2011.

[72] Panagiota Katsikouli, Diego Madariaga, Aline Carneiro Viana, Alberto Tarable, and Marco Fiore. Ductiloc: Energy-efficient location sampling with configurable accuracy. *IEEE Access*, 11:15375–15389, 2023.

[73] Samira Kolyaie, Marjan Yaghooti, and Gilda Majidi. Analysis and simulation of wireless signal propagation applying geostatistical interpolation techniques. *Archiwum Fotogrametrii, Kartografii i Teledetekcji*, 22, 2011.

[74] Abdullah Konak. A kriging approach to predicting coverage in wireless networks. *International Journal of Mobile Network Design and Innovation*, 3(2):65–71, 2009.

[75] NIC Chile Research Labs. Pepa ping dataset. github repository. `https://github.com/niclabs/pepa-ping-mmsys21`. [Online; accessed 18-April-2023].

[76] NIC Chile Research Labs. Pepa ping: Measuring qos for mobile internet. `https://niclabs.cl/pepa/`, 2020. [Online; accessed 18-April-2023].

[77] Felipe Lalanne, Nicolás Aguilera, Alvaro Graves, and Javier Bustos. Adkintun mobile: towards using personal and device context in assessing mobile qos. In *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 49–54. IEEE, 2015.

[78] Markus Laner, Philipp Svoboda, Eduard Hasenleithner, and Markus Rupp. Dissecting 3g uplink delay by measuring in an operational hspa network. In *International Conference on Passive and Active Network Measurement*, pages 52–61. Springer, 2011.

[79] Hyo-Jin Lee, Myung-Sup Kim, James W Hong, and Gil-Haeng Lee. Qos parameters to network performance metrics mapping for sla monitoring. *KNOM Review*, 5(2):42–53, 2002.

[80] William CY Lee. Estimate of local average power of a mobile radio signal. *IEEE Transactions on Vehicular Technology*, 34(1):22–27, 1985.

[81] Anirban Lekharu, Satish Kumar, Arijit Sur, and Arnab Sarkar. A qoe aware lstm based bit-rate prediction model for dash video. In *2018 10th International Conference on Communication Systems & Networks (COMSNETS)*, pages 392–395. IEEE, 2018.

[82] Lanlan Li and Tao Ye. Research on throughput prediction of 5g network based on lstm. *Intelligent and Converged Networks*, 3(2):217–227, 2022.

[83] Xia Li and Petri Mähönen. Grid based cooperative spectrum sensing in cognitive networks under correlated shadowing. In *2012 7th International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, pages 350–355. IEEE, 2012.

[84] Zhijing Li, Ana Nika, Xinyi Zhang, Yanzi Zhu, Yuanshun Yao, Ben Y Zhao, and Haitao Zheng. Identifying value in crowdsourced wireless signal measurements. In *Proceedings of the 26th International Conference on World Wide Web*, pages 607–616. International World Wide Web Conferences Steering Committee, 2017.

[85] Xin-Yu Lin, Te-Wei Ho, Cheng-Chung Fang, Zui-Shen Yen, Bey-Jing Yang, and Feipei Lai. A mobile indoor positioning system based on ibeacon technology. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4970–4973. IEEE, 2015.

[86] Yan Liu and Jack YB Lee. An empirical study of throughput prediction in mobile data networks. In *2015 IEEE global communications conference (GLOBECOM)*, pages 1–6. IEEE, 2015.

[87] Pollere LLC. Transport level passive ping network observer. `https://pollere.net/pping.html`. [Online; accessed 18-April-2023].

[88] Chengwen Luo, Long Cheng, Mun Choon Chan, Yu Gu, Jianqiang Li, and Zhong Ming. Pallas: Self-bootstrapping fine-grained passive indoor localization using wifi monitors. *IEEE Transactions on Mobile Computing*, 16(2):466–481, 2016.

[89] Jari Luomala and Ismo Hakala. Effects of temperature and humidity on radio signal strength in outdoor wireless sensor networks. In *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 1247–1255. IEEE, 2015.

[90] Gerui Lv, Qinghua Wu, Weiran Wang, Zhenyu Li, and Gaogang Xie. Lumos: Towards better video streaming qoe through accurate throughput prediction. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, pages 650–659. IEEE, 2022.

[91] Diego Madariaga, Javier Madariaga, Javier Bustos-Jiménez, and Benjamin Bustos. Improving signal-strength aggregation for mobile crowdsourcing scenarios. *Sensors*, 21(4):1084, 2021.

[92] Diego Madariaga, Javier Madariaga, Martín Panza, Javier Bustos-Jiménez, and Benjamin Bustos. Detecting anomalies at a tld name server based on dns traffic predictions. *IEEE Transactions on Network and Service Management*, 18(1):1016–1030, 2021.

[93] Diego Madariaga, Martín Panza, and Javier Bustos-Jimenéz. I'm only unhappy when it rains: Forecasting mobile qos with weather conditions. In *2018 Network Traffic Measurement and Analysis Conference (TMA)*, pages 1–6. IEEE, 2018.

[94] Diego Madariaga, Martín Panza, and Javier Bustos-Jiménez. Dns traffic forecasting using deep neural networks. In *Machine Learning for Networking: First International Conference, MLN 2018, Paris, France, November 27–29, 2018, Revised Selected Papers 1*, pages 181–192. Springer, 2019.

[95] Diego Madariaga, Lucas Torrealba, Javier Madariaga, Javiera Bermúdez, and Javier Bustos-Jiménez. Analyzing the adoption of quic from a mobile development perspective. In *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*, pages 35–41, 2020.

[96] Diego Madariaga, Lucas Torrealba, Javier Madariaga, Javier Bustos-Jiménez, and Benjamin Bustos. Pepa ping dataset: Comprehensive contextualization of periodic passive ping in wireless networks. In *Proceedings of the 12th ACM Multimedia Systems Conference*, pages 274–280, 2021.

[97] Jaymin D Mankowitz and Andrew J Paverd. Mobile device-based cellular network coverage analysis using crowd sourcing. In *2011 IEEE EUROCON-International Conference on Computer as a Tool*, pages 1–6. IEEE, 2011.

[98] Sue Marek. Chile: Mobile network experience report. march 2020. `https://www.opensignal.com/reports/2020/03/chile/mobile-network-experience`. [Online; accessed 18-April-2023].

[99] Mahesh K Marina, Valentin Radu, and Konstantinos Balampekos. Impact of indoor-outdoor context on crowdsourcing based mobile coverage analysis. In *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, pages 45–50. ACM, 2015.

[100] H Martin, Anthony McGregor, and J Cleary. Analysis of internet delay times. In *Proceedings of Passive and Active Measurment Workshop (PAM)*, 2000.

[101] Lars M Mikkelsen, Steffen R Thomsen, Michael S Pedersen, and Tatiana K Madsen. Netmap-creating a map of application layer qos metrics of mobile networks using crowd sourcing. In *International Conference on Next Generation Wired/Wireless Networking*, pages 544–555. Springer, 2014.

[102] Ondrej Mikle, Karel Slanỳ, Ján Veselỳ, Tomáš Janoušek, and Ondrej Surỳ. Detecting hidden anomalies in dns communication. *Sponsoring Institutions*, page 93, 2011.

[103] Konstantin Miller, Abdel-Karim Al-Tamimi, and Adam Wolisz. Qoe-based low-delay live streaming using throughput predictions. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 13(1):1–24, 2016.

[104] Dimitar Minovski, Niclas Ogren, Christer Ahlund, and Karan Mitra. Throughput prediction using machine learning in lte and 5g networks. *IEEE Transactions on Mobile Computing*, 2021.

[105] Mariyam Mirza, Joel Sommers, Paul Barford, and Xiaojin Zhu. A machine learning approach to tcp throughput prediction. *ACM SIGMETRICS Performance Evaluation Review*, 35(1):97–108, 2007.

[106] Stephen K Mitchell. Comment on "linear versus logarithmic averaging". *The Journal of the Acoustical Society of America*, 41(4A):863–864, 1967.

[107] Dahunsi Folasade Mojisola and Kolawole Gbolahan. Participatory analysis of cellular network quality of service. *International Journal of Computing & ICT Research*, 9(1), 2015.

[108] Ricky KP Mok, Edmond WW Chan, and Rocky KC Chang. Measuring the quality of experience of http video streaming. In *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, pages 485–492. IEEE, 2011.

[109] Massimiliano Molinari, Mah-Rukh Fida, Mahesh K Marina, and Antonio Pescape. Spatial interpolation based cellular coverage prediction with crowdsourced measurements. In *Proceedings of the 2015 ACM SIGCOMM Workshop on Crowdsourcing and Crowdsharing of Big (Internet) Data*, pages 33–38. ACM, 2015.

[110] A Morton. Active and passive metrics and methods (with hybrid types in-between). *Internet Engineering Task Force*, 2016.

[111] S Murray. Beware of spectrum analyzer power averaging techniques. *Microwaves RF*, 45(12):57–66, 2006.

[112] Yasuo Musashi, Masaya Kumagai, Shinichiro Kubota, and Kenichi Sugitani. Detection of kaminsky dns cache poisoning attack. In *Intelligent Networks and Intelligent Systems (ICINIS), 2011 4th International Conference on*, pages 121–124. IEEE, 2011.

[113] Mirco Musolesi and Cecilia Mascolo. Designing mobility models based on social network theory. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(3):59–70, 2007.

[114] Hyeonjun Na, Yongjoo Shin, Dongwon Lee, and Joohyun Lee. Lstm-based throughput prediction for lte networks. *ICT Express*, 2021.

[115] Arvind Narayanan, Eman Ramadan, Rishabh Mehta, Xinyue Hu, Qingxu Liu, Rostand AK Fezeu, Udhaya Kumar Dayalan, Saurabh Verma, Peiqi Ji, Tao Li, et al. Lumos5g: Mapping and predicting commercial mmwave 5g throughput. In *Proceedings of the ACM Internet Measurement Conference*, pages 176–193, 2020.

[116] Joseph Kee-Yin Ng, Kam-Yiu Lam, Quan Jia Cheng, and Kevin Chin Yiu Shum. An effective signal strength-based wireless location estimation system for tracking indoor mobile users. *Journal of Computer and System Sciences*, 79(7):1005–1016, 2013.

[117] NIC Chile. .cl nameservers map. `https://www.nic.cl/estadisticas/mapaDNS.html`. [Online; accessed 18-April-2023].

[118] NIC Chile. Official registry for the .cl cctld. `https://www.nic.cl/`. [Online; accessed 18-April-2023].

[119] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*. SIAM, 1992.

[120] Ashkan Nikravesh, David R Choffnes, Ethan Katz-Bassett, Z Morley Mao, and Matt Welsh. Mobile network performance from user devices: A longitudinal, multidimensional analysis. In *International Conference on Passive and Active Network Measurement*, pages 12–22. Springer, 2014.

[121] Ikponmwosa Oghogho, Fredrick O Edeko, and Joy Emagbetere. Measurement and modelling of tcp downstream throughput dependence on snr in an ieee802. 11b wlan system. *Journal of King Saud University-Engineering Sciences*, 30(2):170–176, 2018.

[122] Eduardo Mucelli Rezende Oliveira, Aline Carneiro Viana, Carlos Sarraute, Jorge Brea, and Ignacio Alvarez-Hamelin. On the regularity of human mobility. *Pervasive and Mobile Computing*, 33:73–90, 2016.

[123] Olasunkanmi F Oseni, Segun I Popoola, Robert O Abolade, and Oluwole A Adegbola. Comparative analysis of received signal strength prediction models for radio network planning of gsm 900 mhz in ilorin, nigeria. *International Journal of Innovative Technology and Exploring Engineering*, 4(3):45–50, 2014.

[124] Olav Østerbø. Scheduling and capacity estimation in lte. In *2011 23rd International Teletraffic Congress (ITC)*, pages 63–70. IEEE, 2011.

[125] Imane Oussakel, Philippe Owezarski, and Pascal Berthou. Cellular uplink bandwidth prediction based on radio measurements. In *Proceedings of the 17th ACM International Symposium on Mobility Management and Wireless Access*, pages 111–118, 2019.

[126] Martin Panza, Diego Madariaga, and Javier Bustos-Jimenez. Extracting human behavior patterns from dns traffic. *Annals of Telecommunications*, 77(5-6):407–420, 2022.

[127] JD Parsons, MF Ibrahim, and RJ Samuel. Median signal strength prediction for mobile radio propagation in london. *Electronics Letters*, 16(5):172–173, 1980.

[128] Segun I Popoola, Aderemi A Atayero, and Nasir Faruk. Received signal strength and local terrain profile data for radio network planning and optimization at gsm frequency bands. *Data in brief*, 16:972–981, 2018.

[129] J Portelinha, F Martins, and P Cardieri. Effects of correlated shadowing on cooperative spectrum sensing. In *International Workshop on Telecommunications (IWT)*, 2013.

[130] Leonid Portnoy, Eleazar Eskin, and Sal Stolfo. Intrusion detection with unlabeled data using clustering. In *In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001*. Citeseer, 2001.

[131] Jon Postel. Transmission control protocol. STD 7, RFC Editor, September 1981. http://www.rfc-editor.org/rfc/rfc793.txt.

[132] Luc Pronzato. Minimax and maximin space-filling designs: some properties and methods for construction. *Journal de la Societe Française de Statistique*, 158(1):7–36, 2017.

[133] Jing Qiao. .nz dns traffic: Trend and anomalies. https://blog.nzrs.net.nz/nz-dns-traffic-trend-and-anomalies/, 2017. 2020-04-30.

[134] Darijo Raca, Jason J Quinlan, Ahmed H Zahran, and Cormac J Sreenan. Beyond throughput: A 4g lte dataset with channel and context metrics. In *Proceedings of the 9th ACM multimedia systems conference*, pages 460–465, 2018.

[135] Darijo Raca, Ahmed H Zahran, Cormac J Sreenan, Rakesh K Sinha, Emir Halepovic, Rittwik Jana, and Vijay Gopalakrishnan. On leveraging machine and deep learning for throughput prediction in cellular networks: Design, performance, and challenges. *IEEE Communications Magazine*, 58(3):11–17, 2020.

[136] Darijo Raca, Ahmed H Zahran, Cormac J Sreenan, Rakesh K Sinha, Emir Halepovic, Rittwik Jana, Vijay Gopalakrishnan, Balagangadhar Bathula, and Matteo Varvello. Incorporating prediction into adaptive streaming algorithms: a qoe perspective. In *Proceedings of the 28th ACM SIGMM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 49–54, 2018.

[137] Darijo Raca, Ahmed H Zahran, Cormac J Sreenan, Rakesh K Sinha, Emir Halepovic, Rittwik Jana, Vijay Gopalakrishnan, Balagangadhar Bathula, and Matteo Varvello. Empowering video players in cellular: Throughput prediction from radio network measurements. In *Proceedings of the 10th ACM Multimedia Systems Conference*, pages 201–212, 2019.

[138] Vaclav Raida, Philipp Svoboda, Martin Lerch, and Markus Rupp. Crowdsensed performance benchmarking of mobile networks. *IEEE Access*, 7:154899–154911, 2019.

[139] Vaclav Raida, Philipp Svoboda, and Markus Rupp. Real world performance of lte downlink in a static dense urban scenario-an open dataset. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pages 1–6. IEEE, 2020.

[140] Anderson R Ramos, Bruno C Silva, Marisa S Lourenço, Emanuel B Teixeira, and Fernando J Velez. Mapping between average sinr and supported throughput in 5g new radio small cell networks. In *2019 22nd International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 1–6. IEEE, 2019.

[141] Abbas Razaghpanah, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Christian Kreibich, Phillipa Gill, Mark Allman, and Vern Paxson. Haystack: In situ mobile traffic analysis in user space. *arXiv preprint arXiv:1510.01419*, pages 1–13, 2015.

[142] G Ricker and J Williams. Averaging logarithms for detection and estimation (corresp.). *IEEE Transactions on Information Theory*, 20(3):378–382, 1974.

[143] Pieter Robberechts, Maarten Bosteels, Jesse Davis, and Wannes Meert. Query log analysis: Detecting anomalies in dns traffic at a tld resolver. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 55–67. Springer, 2018.

[144] Daniel Romero, Seung-Jun Kim, Georgios B Giannakis, and Roberto López-Valcarce. Learning power spectrum maps from quantized power measurements. *IEEE Transactions on Signal Processing*, 65(10):2547–2560, 2017.

[145] Peter Romirer-Maierhofer, Fabio Ricciato, Alessandro D'Alconzo, Robert Franzan, and Wolfgang Karner. Network-wide measurements of tcp rtt in 3g. In *International Workshop on Traffic Monitoring and Analysis*, pages 17–25. Springer, 2009.

[146] Sanae Rosen, Sung-ju Lee, Jeongkeun Lee, Paul Congdon, Z Morley Mao, and Ken Burden. Mcnet: Crowdsourcing wireless performance measurements through the eyes of mobile devices. *IEEE Communications Magazine*, 52(10):86–91, 2014.

[147] Sebin Sabu, S Renimol, D Abhiram, and B Premlet. Effect of rainfall on cellular signal strength: A study on the variation of rssi at user end of smartphone during rainfall. In *2017 IEEE Region 10 Symposium (TENSYMP)*, pages 1–4. IEEE, 2017.

[148] Alassane Samba, Yann Busnel, Alberto Blanc, Philippe Dooze, and Gwendal Simon. Instantaneous throughput prediction in cellular networks: Which information is needed? In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 624–627. IEEE, 2017.

[149] Irene Santos, Juan José Murillo-Fuentes, and Petar M Djurić. Recursive estimation of dynamic rss fields based on crowdsourcing and gaussian processes. *IEEE Transactions on Signal Processing*, 67(5):1152–1162, 2019.

[150] Aaron Schulman, Vishnu Navda, Ramachandran Ramjee, Neil Spring, Pralhad Deshpande, Calvin Grunewald, Kamal Jain, and Venkata N Padmanabhan. Bartendr: a practical approach to energy-aware cellular data scheduling. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 85–96. ACM, 2010.

[151] G Nychis V Sekar, DG Anderson, et al. An empirical evaluation of entropy-based anomaly detection. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement,, ACM Press, pp151-156*, 2008.

[152] Muhammad Zubair Shafiq, Lusheng Ji, Alex X Liu, Jeffrey Pang, Shobha Venkataraman, and Jia Wang. A first look at cellular network performance during crowded events. *ACM SIGMETRICS Performance Evaluation Review*, 41(1):17–28, 2013.

[153] Hyojeong Shin, Yohan Chon, Yungeun Kim, and Hojung Cha. Mri: Model-based radio interpolation for indoor war-walking. *IEEE Transactions on Mobile Computing*, 14(6):1231–1244, 2014.

[154] Yihang Song and Urs Hengartner. Privacyguard: A vpn-based platform to detect information leakage on android devices. In *Proceedings of the 5th Annual ACM CCS*

Workshop on Security and Privacy in Smartphones and Mobile Devices, pages 15–26, 2015.

[155] Ain A Sonin. The physical basis of dimensional analysis. *Department of Mechanical Engineering, MIT, Cambridge, MA*, pages 1–57, 2001.

[156] Sebastian Sonntag, Jukka Manner, and Lennart Schulte. Netradar-measuring the wireless world. In *2013 11th International Symposium and Workshops on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, pages 29–34. IEEE, 2013.

[157] Sebastian Sonntag, Lennart Schulte, and Jukka Manner. Mobile network measurements-it's not all about signal strength. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 4624–4629. IEEE, 2013.

[158] Beatriz Soret, Preben Mogensen, Klaus I Pedersen, and Mari Carmen Aguayo-Torres. Fundamental tradeoffs among reliability, latency and throughput in cellular networks. In *2014 IEEE Globecom Workshops (GC Wkshps)*, pages 1391–1396. IEEE, 2014.

[159] StatCounter. Global stats: Desktop vs mobile market share worldwide. `https://gs.statcounter.com/platform-market-share/desktop-mobile/worldwide/`. [Online; accessed 18-April-2023].

[160] Stephen D Strowes. Passively measuring tcp round-trip times. *Communications of the ACM*, 56(10):57–64, 2013.

[161] Keen Sung, Joydeep Biswas, Erik Learned-Miller, Brian N Levine, and Marc Liberatore. Server-side traffic analysis reveals mobile location information over the internet. *IEEE Transactions on Mobile Computing*, 18(6):1407–1418, 2018.

[162] Jukka Talvitie and Elena Simona Lohan. Modeling received signal strength measurements for cellular network based positioning. In *2013 International Conference on Localization and GNSS (ICL-GNSS)*, pages 1–6. IEEE, 2013.

[163] TeleGeography. Submarine cable map: Chile. `https://www.submarinecablemap.com/#/country/chile`. [Online; accessed 18-April-2023].

[164] Francesco Vacirca, Fabio Ricciato, and René Pilz. Large-scale rtt measurements from an operational umts/gprs network. In *First International Conference on Wireless Internet (WICON'05)*, pages 190–197. IEEE, 2005.

[165] Reinaldo A Valenzuela, Orlando Landron, and DL Jacobs. Estimating local mean signal strength of indoor multipath propagation. *IEEE transactions on vehicular technology*, 46(1):203–212, 1997.

[166] Ricardo Villamarín-Salomón and José Carlos Brustoloni. Identifying botnets using anomaly detection techniques applied to dns traffic. In *2008 5th IEEE Consumer Communications and Networking Conference*, pages 476–481. IEEE, 2008.

[167] Wenjia Wang, Rui Tuo, and CF Jeff Wu. On prediction properties of kriging: Uniform error bounds and robustness. *Journal of the American Statistical Association*, pages 1–27, 2019.

[168] Bo Wei, Wataru Kawakami, Kenji Kanai, Jiro Katto, and Shangguang Wang. Trust: A tcp throughput prediction method in mobile networks. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2018.

[169] Bo Wei, Hang Song, Shangguang Wang, Kenji Kanai, and Jiro Katto. Evaluation of throughput prediction for adaptive bitrate control using trace-based emulation. *IEEE Access*, 7:51346–51356, 2019.

[170] Holger Wendland. *Scattered data approximation*, volume 17. Cambridge university press, 2004.

[171] Li Wenwei, Zhang Dafang, Yang Jinmin, and Xie Gaogang. On evaluating the differences of tcp and icmp in network measurement. *Computer Communications*, 30(2):428–439, 2007.

[172] Z Whang and Shian-Shyong Tseng. Anomaly detection of domain name system (dns) query traffic at top level domain servers. *Scientific Research and Essays*, 6(18):3858–3872, 2011.

[173] Daniel Wong and Donald C Cox. Estimating local mean signal power level in a rayleigh fading environment. *IEEE transactions on vehicular technology*, 48(3):956–959, 1999.

[174] Chris Wormald. Predicting user availability from aggregated signal strength data, 2013. US Patent 8,396,470.

[175] Fengli Xu, Yuyun Lin, Jiaxin Huang, Di Wu, Hongzhi Shi, Jeungeun Song, and Yong Li. Big data driven mobile traffic understanding and forecasting: A time series approach. *IEEE transactions on services computing*, 9(5):796–805, 2016.

[176] Qiang Xu, Sanjeev Mehrotra, Zhuoqing Mao, and Jin Li. Proteus: network performance forecast for real-time, interactive mobile applications. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 347–360, 2013.

[177] Sandeep Yadav, Ashwath Kumar Krishna Reddy, AL Narasimha Reddy, and Supranamaya Ranjan. Detecting algorithmically generated domain-flux attacks with dns traffic analysis. *IEEE/Acm Transactions on Networking*, 20(5):1663–1677, 2012.

[178] Chaoqun Yue, Ruofan Jin, Kyoungwon Suh, Yanyuan Qin, Bing Wang, and Wei Wei. Linkforecast: Cellular link bandwidth prediction in lte networks. *IEEE Transactions on Mobile Computing*, 17(7):1582–1594, 2017.

[179] Ahmed Hamdy Zahran, Darijo Raca, and Cormac J Sreenan. Arbiter+: Adaptive rate-based intelligent http streaming algorithm for mobile networks. *IEEE Transactions on Mobile Computing*, 17(12):2716–2728, 2018.

[180] Bojan Zdrnja, Nevil Brownlee, and Duane Wessels. Passive monitoring of dns anomalies. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 129–139. Springer, 2007.

[181] Roman Zhohov, Alexandros Palaios, and Philipp Geuer. One step further: Tunable and explainable throughput prediction based on large-scale commercial networks. In *2021 IEEE 4th 5G World Forum (5GWF)*, pages 430–435. IEEE, 2021.

Annex A

PePa Ping dataset

In order to support the usefulness of our proposed measurement methodology (Chapter 3), we performed different analyses that evidenced the high value of the data collected by our passive measurement approach.

As both datasets were collected by several volunteers running our mobile measurement tool, we can thoroughly analyze the data to obtain information that extensively characterize the users in terms of Internet usage, network access, and mobility. However, it is important to note that as the users were recruited following a convenience sampling process, the collected datasets are influenced by the over-representation of university students in the population sample.

Even when the outcomes of these analyses are specific to the measured population, our measurement methodology can be deployed in different scenarios to study any other population of interest, independent of their geographic location or size. In each case, the dataset collected by a set of mobile users could provide a broad picture of the target population being studied.

In the following, we employ the data collected by our mobile measurement tool to analyze the adoption of different network protocols from the perspective of mobile users

A.1 Protocol Analysis

A.1.1 DNS protocol

During the last decade, three major protocols have been standardized for secure DNS transport: DNS over TLS (DoT) in 2016 [58], DNS over HTTPS (DoH) in 2018 [57], and DNS over QUIC (DoQ) in 2022 [60]. These newer security protocols, in addition to the classical versions of the DNS protocol (unencrypted DNS traffic over UDP or TCP), encompass the different alternatives in which we can find DNS traffic in the wild.

To analyze the adoption of these protocols, we filtered the DNS traffic from both datasets

according to the following rules:

1. DNS over UDP (**DoUDP**): All network flows connected to UDP port 53.
2. DNS over TCP (**DoTCP**): All network flows connected to TCP port 53.
3. DNS over TLS (**DoT**): All network flows connected to TCP port 853.
4. DNS over HTTPS (**DoH**): All network flows connected to TCP port 443, whose destination IP addresses appeared in the first 3 categories.
5. DNS over QUIC (**DoQ**): All network flows connected to UDP port 443, whose destination IP addresses appeared in the first 3 categories. Additionally, all network flows connected to UDP port 853.

However, it is important to mention that the following insights do not include information about DNS relying on TCP port 53 (DoTCP), as the number of DoTCP network flows were remarkably low (less than 0.1% of the total of DNS network flows).

Figure A.1 shows the percentage of network flows related to each type of DNS protocol. In both years, more than 80% of the DNS flows used the classical DNS over UDP version, followed by DNS flows using the DoT protocol accounting for nearly 10% of the DNS connections in both datasets. Therefore, during those periods of time, the vast majority of DNS connections were established without encryption.

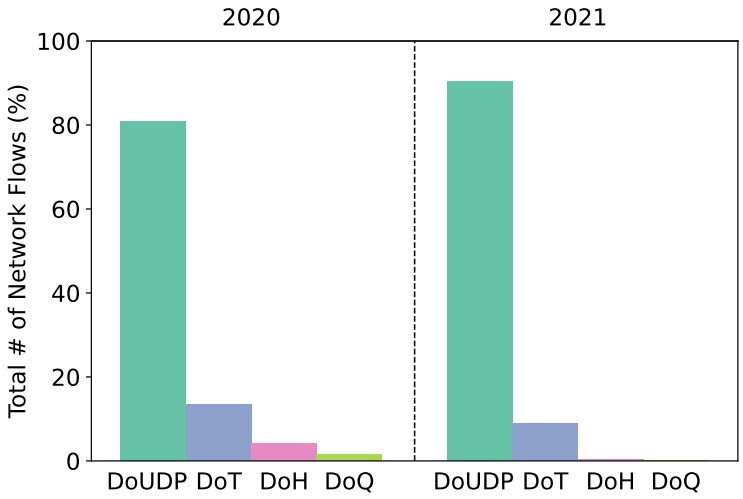


Figure A.1: Percentage of DNS flows related to each type of DNS protocol.

Similarly, Figure A.2 shows the percentage of the total DNS traffic related to each type of DNS protocol (considering both transmitted and received bytes). In this case, most of the DNS traffic is referable to encrypted connections, particularly to DoT. This inequality between the number of connections and the amount of network traffic can be understood by the nature of each protocol.

Indeed, Figure A.3 illustrates the distribution of total traffic (transmitted and received) for all the types of DNS protocols in the 2021 dataset. The higher amount of network

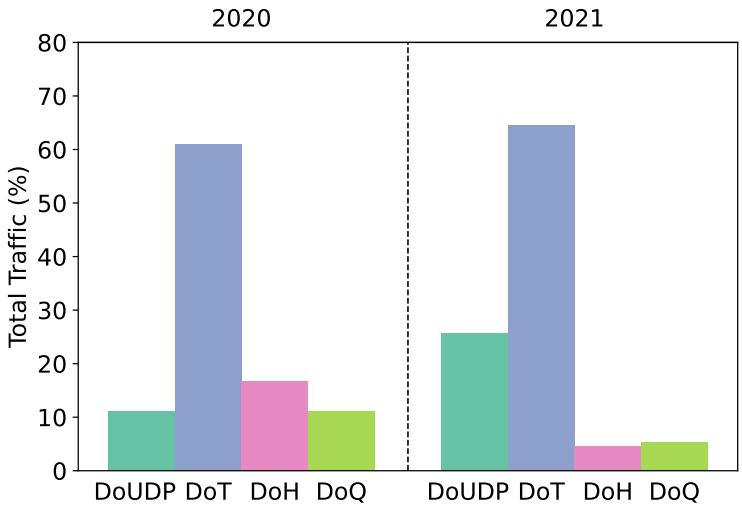


Figure A.2: Percentage of DNS traffic related to each type of DNS protocol.

traffic related to encrypted DNS connections can be mainly explained by two factors. Firstly, encrypted DNS protocols are expected to receive multiple DNS queries/responses over a single session, and therefore, contrarily to the single DNS query/response behaviour of DoUDP connections. Secondly, and most important, encrypted DNS connections have an intrinsic traffic overhead related to the establishment of secure sessions.

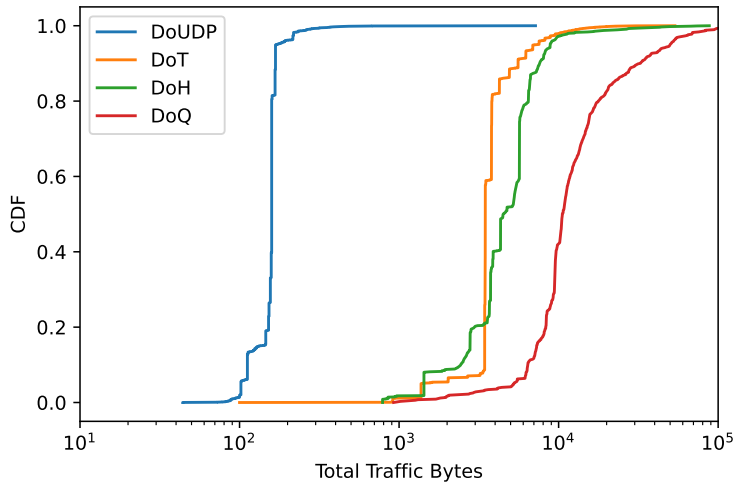


Figure A.3: Distribution of network traffic for connections relying on different types of DNS protocols in the 2021 dataset.

Regarding the overhead related to the establishment of secure sessions, the data collected by our measurement approach can be used for a deeper study. Figure A.4 shows the number of transmitted and received bytes for all DoT connections in the 2021 dataset. The figure clearly evidences two different clusters (labeled as Class 1 and Class 2) which present different traffic behavior. Class 1 accounts for the higher number of DoT connections (92%), where even the connections with lower traffic have a remarkable overhead on the number of received bytes. This overhead can be explained by the certificate of the server, which is sent to the client during the TLS handshake. Differently, DoT connections from Class 2 do not appear

to have such overhead. Therefore, Class 2 can be related to connections that resume an existing TLS session, and therefore, the sending of the server certificate is omitted by the TLS handshake. For both classes, the proportional growth between transmitted and received bytes can be related to the exchange of multiple DNS queries/responses over the same session.

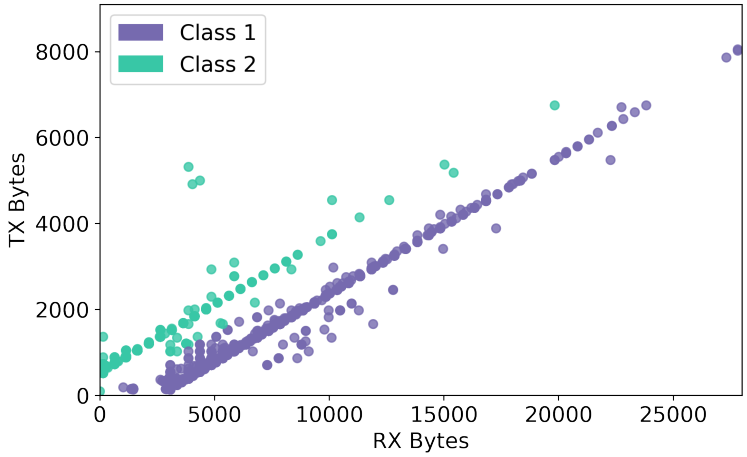


Figure A.4: Relation between transmitted and received bytes for DoT connections in the 2021 dataset.

The DNS resolver employed by user devices is usually provided by their ISP by default, and therefore, the DNS resolution time can be used to evaluate ISPs quality of service as perceived by end users. From the collected environmental information, we can determine whether each network flow used a mobile network or a WiFi network to establish a connection to the destination IP address. For the case of network flows using a mobile network, we can assign them to a specific mobile ISP according to the information related to each user's SIM card. For the case of network flows using a WiFi network, we can assign them to a specific ISP according to the organization managing the autonomous system to which the sending router belongs.

Thus, we analyzed the performance of DNS according to the two types of DNS protocols accounting for the highest number of connections: DoUDP and DoT. As DoUDP flows contain a single query/response exchange, we can directly compute the DNS resolution time as the duration of each DoUDP connection. In contrast, as DoT allows the exchange of multiple DNS queries/responses over the same session, we evaluate the performance of each DoT connection by its RTT value, obtained by our PePa Ping measurement approach.

Figure A.5 shows the performance of DoUDP and DoT flows corresponding to the three mobile ISPs with the highest presence in the 2021 dataset. This analysis is important as it exhibits no direct correlation between DoUDP and DoT performance at the mobile ISP level.

Similarly, Figure A.6 shows the performance of DoUDP and DoT flows corresponding to the three home ISPs with the highest presence in the 2021 dataset. As for mobile ISPs, the figure does not show a clear correlation between DoUDP and DoT performance at the ISP level.

As shown before, the environmental information collected by our monitoring tool is not only useful for analyzing the adoption of network protocols from the perspective of mobile

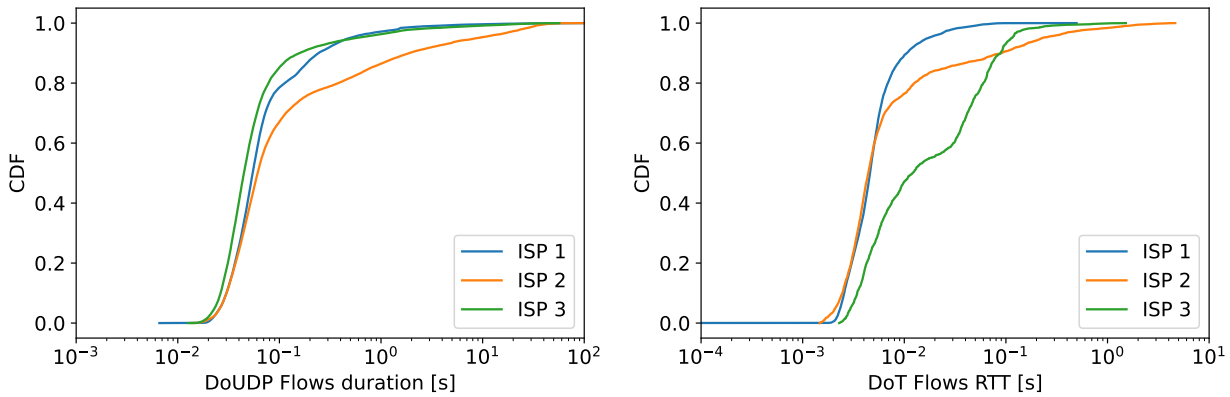


Figure A.5: Performance of DNS flows for mobile ISPs.

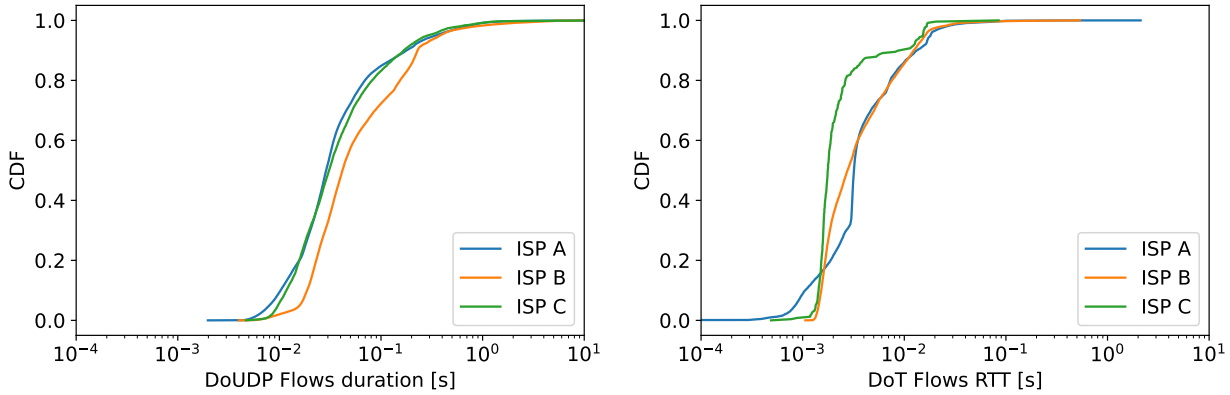


Figure A.6: Performance of DNS flows for home ISPs.

users, but also for evaluating the quality of services provided by ISPs.

A.1.2 QUIC protocol

First introduced in 2013 and standardized in 2021 [63], QUIC protocol has been in constant development and it has gained great importance over time. Indeed, several technology companies have put great efforts into adopting QUIC protocol for the delivery of their content. Therefore, the data collected by our measurement tool can be useful to analyze the adoption of QUIC from the perspective of mobile devices' traffic.

To analyze the adoption of QUIC protocol, we filtered the web traffic from both datasets according to the following rules:

1. **QUIC**: All network flows connected to UDP port 443.
2. **HTTP**: All network flows connected to TCP port 80.
3. **HTTPS**: All network flows connected to TCP port 443.

At analyzing the 2020 dataset, we found 173 different Android apps using QUIC, from a total of 831 apps analyzed (20.8%). Similarly, in the 2021 dataset, we found 258 different Android apps using QUIC, from a total of 1254 apps analyzed (20.6%). Thus, between 2020 and 2021, we did not notice a variation on the percentage of Android apps using the QUIC protocol.

As shown in Table A.1, in both datasets the vast majority of QUIC traffic was served by four major organizations: Google, Snap Inc., Uber Technologies, and Facebook (now Meta). In 2020, almost 95% of the total QUIC traffic was served by Google. However, in 2021, the percentege of QUIC traffic served by Google decreased significantly, given the increase in the QUIC traffic served Facebook.

Table A.1: Distribution of QUIC traffic among major organizations

Organization	**2020**	**2021**
Google	94.97%	52.80%
Facebook	5.00%	47.07%
Snap	0.00%	0.04%
Uber	0.03%	0.01%

For these four major organizations, we also analyzed the distribution of web traffic using the different web protocols (HTTP, HTTPS, and QUIC). As shown in Figure A.7 and Figure A.8, we found that Facebook was the organization that mostly increased its QUIC traffic from 2020 to 2021, increasing from 3.07% to 83.3%. This phenomenon is consistent with the Facebook company announcing in 2020 its decision to use QUIC as its de facto protocol [69].

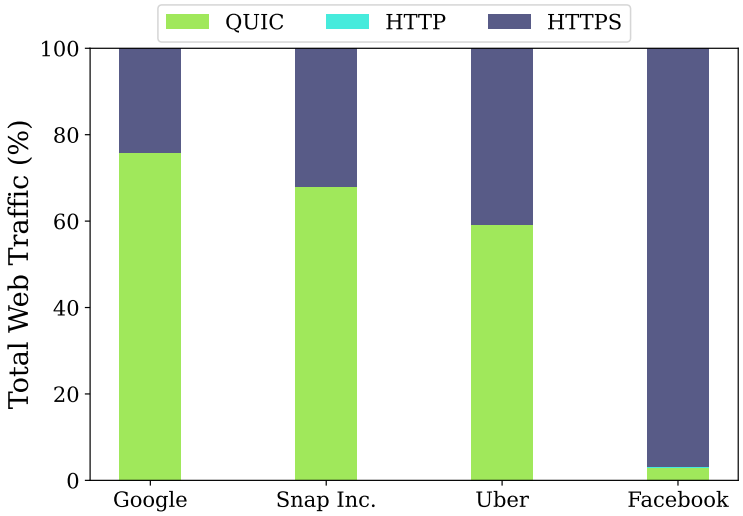


Figure A.7: Distribution of web traffic for major organizations in the 2020 dataset.

In this case, the environmental information collected by our monitoring tool is shown to be essential, as it allows a deep insight about which Android applications use the QUIC protocol for accessing the web, and which organizations are serving those contents.

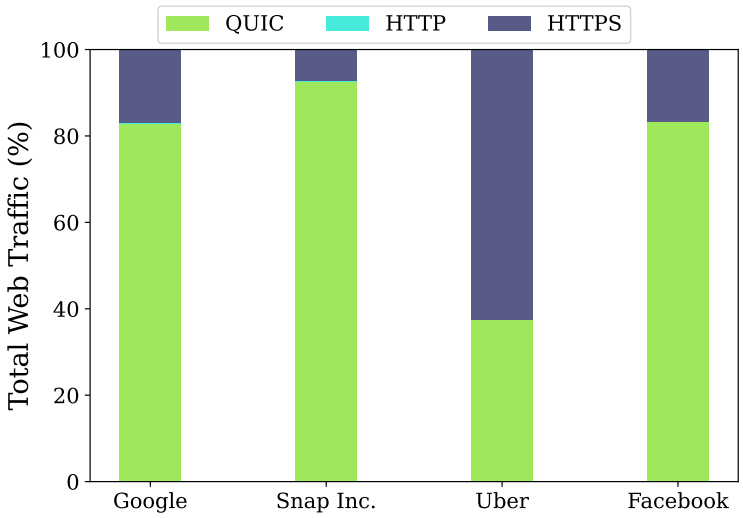


Figure A.8: Distribution of web traffic for major organizations in the 2021 dataset.

Annex B

Proof of Mathematical Expression (5.16)

It is clear that

$$\mathbb{E}\Big[\max_{i\in[1:m]}\big|P(x_i)-I_N(x_i)\big|\Big]\leq\mathbb{E}\Big[\sup_{x\in A}\big|P(x)-I_N(x)\big|\Big] \tag{B.1}$$

Then, Corollary 1 of Wang et al. [167] holds that, selecting $p=1$,

$$\mathbb{E}\Big[\sup_{x\in A}\big|P(x)-I_N(x)\big|\Big]=\mathcal{O}(P_{\Phi,X}\log^{1/2}(1/P_{\Phi,X})),\ \text{as}\ P_{\Phi,X}\to 0$$

It can be noticed that function $f(x)=x\log^{1/2}(1/x)$ satisfies the following limit:

$$\lim_{x\to 0^+} f(x)=0$$

Therefore,

$$\lim_{P_{\Phi,X}\to 0^+}\mathbb{E}\Big[\sup_{x\in A}\big|P(x)-I_N(x)\big|\Big]=0 \tag{B.2}$$

Considering that power measurements $P(\cdot)$ over a rectangle area A can be modeled by Gaussian Processes (hypothesis $(\star)$ of Theorem 2), then Theorem 11.22 of Wendland et al. [170] states that *exist positive constants c and h_0 depending only on A such that $P_{\Phi,X}\leq h_N^{c/h_N}$ provided that $h_N\leq h_0$.*

It can be seen that function $f(x)=x^{c/x}$ satisfies the following limit:

$$\lim_{x\to 0^+} f(x)=0$$

Consequently, it can be verified that

$$\lim_{h_N \to 0^+} h_N^{c/h_N} = 0$$

and taking into account that Theorem 11.22 of Wendland et al. [170] states that $P_{\Phi,X} \leq h_N^{c/h_N}$, we have

$$h_N \to 0^+ \implies P_{\Phi,X} \to 0^+$$

Then, Equation (B.2) can be rewritten in terms of h_N as

$$\lim_{h_N \to 0^+} \mathbb{E}\Big[\sup_{x \in A} \big| P(x) - I_N(x) \big| \Big] = 0 \tag{B.3}$$

Finally, by plugging (B.3) into (B.1) we obtain the desired expression in Equation (5.16):

$$\lim_{h_N \to 0} \mathbb{E}\Big[\max_{i \in [1:m]} \big| P(x_i) - I_N(x_i) \big| \Big] = 0$$