AN ABSTRACT OF THE DISSERTATION OF

Abdelkader Aljerme for the degree of Doctor of Philosophy in Electrical and Computer Engineering presented on March 19, 2020.

Title: Advanced Time-varying Approaches for Modeling the Multipath Channel in Wireless Network

Abstract approved: ____

Huaping Liu

This dissertation proposes the use of advanced time-varying approaches for modeling the dynamics of the multipath channel in wireless communication networks. These advanced time-varying approaches include linear Kalman innovation models in observable block companion form, and neural network-based models. The effectiveness of these type of models is evaluated through three case studies. The first case study involves the identification of a linear time-varying Kalman innovation model, for describing measured received signal strength (RSSI) as a function of the speed of the link in an indoor multipath wireless channel. Results for this first case study show that the model exhibits both accuracy and robustness. The second case study evaluates the suitability of using a linear time-varying

Kalman innovation model of the RSSI, for secret key generation in the physical layer of multipath wireless channels. It was found that the residuals of the Kalman model, due to their significant randomness, exhibit a notable potential for secret key generation; indeed, improved values of maximum channel capacity for secret key generation were achieved. At last, the third case study includes the identification of a neural network-based autoregressive moving average with exogenous inputs (NN-ARMAX) model and of a neural network-based autoregressive with exogenous inputs (NN-ARX) model, for describing traffic in a 4G-LTE network. Both models showed similar performance, but the NN-ARMAX has the advantage that it can be converted to a linear time-varying Kalman innovation model, and thus can be used for the implementation of advanced strategies for controlling the operation of the network.

Advanced Time-varying Approaches for Modeling the Multipath Channel in Wireless Network

by

Abdelkader Aljerme

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented March 19, 2020
Commencement June 2020

Doctor of Philosophy dissertation of Abdelkader Aljerme presented on March 19, 2020.

APPROVED:

Major Professor, representing Electrical and Computer Engineering

---

Head of the School of Electrical Engineering and Computer Scinece

---

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Abdelkader Aljerme, Author

ACKNOWLEDGEMENTS

My doctoral journey at Oregon State University has been an extremely rewarding experience. There are many people that I would like to acknowledge.

First, I would like to express my sincere gratitude and appreciation to my academic advisors Professor in EECS at Oregon State University Huaping Liu. I would like to thank him for offering me the chance to join his group and work under his supervision. His expertise, help, valuable suggestions, continuous support, and the fruitful discussions that we had throughout this journey added considerably to my graduate experience and my ability to carry out scientific research at OSU. My advisor's help was crucial in achieving this work and for that, I am forever grateful. I would like to express my sincere gratitude also to my committee members Professor Bello Bose, Professor Eduardo Cattilla Sanchez, Professor Tejasvi Anand, and professor David Porter for accepting to serve as members of the committee of my defense and for the time they devoted to read this thesis. I would like to deeply thank my dear and beloved parents, Amina Taher and Ahmed Aljerme, and would like to dedicate this thesis to them with my utmost appratieation and praying for them, for their unlimited support, love and encouragements throughout all the endeavors that I have been through before they passed away. I would like also to thank and dedicate this thesis to

my beloved and faithful wife Munira Alahrish for her patience, love and constant valuable support during this very important step of our live. Finally a special dedication, and love to my beloved kids Omar, Zaid, Lubna, Lena, Sara, Selma to whom I wish happiness and success in their lives.

CONTRIBUTION OF AUTHORS

This dissertation proposes the use of advanced time-varying approaches for modeling the dynamics of the multipath channel in wireless communication networks. These advanced time-varying approaches include the linear Kalman innovation model in observable block companion form, and two neural network-based models, namely the neural network-based autoregressive moving average with exogenous inputs (NN-ARMAX) model and the neural network-based autoregressive with exogenous inputs (NN-ARX) model.

Previous published works that apply Kalman models for channel estimation [1–4] and key generation [5–9] in wireless networks use the classical Kalman filter. Compared to such classical Kalman approach, the Kalman innovation model used in this work has the advantage that it does not require to estimate the properties of the noise, while maintaining the typical estimation accuracy and time-varying features.

On the other hand, the neural network-based models used here outperform other black box approaches using simple models such as double Gaussian [10] and trapezoidal [11]. In addition, compared to previous works that use neural networks approaches for modeling the dynamics of wireless networks [12–46], the NN-ARX and NN-ARMAX models trained using OS-ELM exhibit an outstanding accuracy and significantly greater training speed; the latter makes them very suitable for identifying models for describing time-varying dynamics of different phenomena in wireless communication networks.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

Page

LIST OF FIGURES

LIST OF TABLES

Chapter 1: OVERVIEW

1.1 Introduction

Because of the significant worldwide growth in the number of users of wireless networks in the previous two decades, two main challenges have arisen in the management of such networks: achieve an optimal operation and guarantee secure communications for the users.

Optimal operation involves balancing two conflicting objectives, namely quality and coverage. In particular, when the coverage area of a station or group of stations is wider they serve more users, but the access speed (one of parameters that define the quality of experience (QoE) perceived by users) tends to reduce for all these users. In contrast, when such coverage area is narrower, the access speed is higher thus increasing the QoE, but there is a smaller number of clients satisfied. Then, the goal becomes offering a sufficiently high access speed to the greater number of users. In other words, it is desired to optimize the network in order to satisfy the demand, with a minimum required quality and the minimum amount of resources [1] [51].

Successfully solving this optimization task is a significant challenge due to the heterogeneity of current networks, because of the coexistence of different technologies, namely

5G, 4G LTE, WCDMA (3^{RD} generation) and GSM (2^{ND} generation). Besides, actual networks are more heterogeneous (HetNet) because of the incorporation of WiFi networks, various types of cells such as pico and femto cells [51], and even sensor networks that often need to be secure [52–56].

Most mobile operators rely on experience and intuition to tackle the optimal operation issue, but this strategy does not guarantee good results. A better choice might be to apply more rigorous optimization strategies, based on mathematical models that accurately describe the dynamics of the channel.

On the other hand, with respect to security, the shared nature of the wireless channel provides a natural eavesdropping and intervention possibility to unintended users. Anyone with a tuned receiver within a certain radius that has an adequate signal-to-interference-plus-noise ratio (SINR) may eavesdrop [57]. It becomes critical to implement effective encrypting mechanisms to guarantee secure communications between every pair of legitimate users [47].

Cryptographic techniques demand the exchange of encrypted keys at one point during the encryption. Traditional security mechanisms are mainly based on the generation, distribution and renewal of shared secret keys in real time, for every active wireless link in the network. This is a nontrivial task in large wireless networks, because of mobility and scalability issues [58].

Currently, wireless security relies on cryptographic techniques and protocols that lie at the upper layers of the network [58]. These techniques require computationally demanding and mathematically complex key management schemes, and have limited capacity for key generation. A promising alternative might be to generate secret keys in the physical layer (channel), exploiting the randomness produced by the multipath, which is time-varying and unique for every user, and the Doppler effect of moving receivers [47]. A review of methods for secret key generation in the physical layer is presented in [58] and [5]. Since the capacity of generating secure keys is proportional to the randomness of the channel, in theory a channel with multipath has an infinite capacity for generating keys. However, in practice computational issues limit this capacity.

Such randomness is present in channel variables such as the gain, the received signal strength (RSS), the angle of arrival (AoA), and the distance between the legitimate users. In addition, variables suitable to be used for key generation (i) must be measurable by the communicating parties, denoted as Alice and Bob, to guarantee they both can generate the same key, (ii) must not be measurable by other malicious parties, to assure that the key is completely secret, and (iii) should be independent of previous or future measurements, to increase overall security [5]. The gain is commonly employed because both users can easily measure it. Now, the practical implementation of a key generation mechanism using any variable that fulfills the aforementioned conditions requires finding a way to capture its randomness.

From the discussion above, it can be stated that a model of the dynamics of the channel becomes a rather critical tool to effectively tackle the aforementioned challenges. The channel transforms the transmitted signal into the received signal. Characterizing the channel in a wireless communication system, also known as channel estimation, is defined as the process of characterizing the effect of the physical channel on the input sequence [6]. Channel estimation is currently an active research topic [48].

Now, creating models of the channel based on physical principles is difficult because of the variety of services offered by the network, namely internet access, text messages and calls. Black-box models identified from data become an attractive alternative to physical principles models, especially considering the significant amount of measured data typically available in mobile telephony and wireless networks, and the storage capacity and computing power of actual computers [51].

Yazti and Krishnaswamy categorize the data from mobile networks in two groups, namely network-level and app-level ([12], [13]). The app-level is data generated by specific applications in the network terminal elements; this data enables to closely investigate the profiles of the subscriber, but requires employing more resources (energy and radio spectrum resources, among others) and the permission from the user. On the other hand, the network-level is data coming from network integrated elements (routers, switches, nodes, among others); this data can be obtained relatively easier and without the permission from the user, but does not provide information about the consumption's profile [51].

Indeed, network-level data is the most frequently utilized when optimizing radio access networks (RAN). Nevertheless, having some information about the profile of the user associated to each node of the network is very useful, when it comes to finding the configuration that generates greater profits, while guaranteeing a quality that satisfies the users.

Among the methods that use network-level data, the best for estimating channel parameters is known as channel state indicator (CSI), which consists of determining a complex transfer function of the channel through the frequency response (CFR) ([1–3, 57, 59]) or impulse response (CIR) ([3, 4, 59]). However, this method is computationally expensive.

As an alternative, the received signal strength indicator (RSSI) offers a much smaller computational cost, at the expense of a reduced capacity. The RSSI is related to the power of the signal and enables to distinguish one channel with multipath from another [48], and has become the more common method for channel measurement, being available in most of the actual transceivers ([1, 60–64]).

Motivated by the well-demonstrated ability of Kalman filters to build models of systems from noisy measurements, standard Kalman models have been applied for channel estimation ([1–4])

and key generation ([5, 7–9, 65]) using RSSI measurements. McGuire [5] used a standard Kalman filter to estimate the dynamics of the channel, and found that the method is effective but more computational efficiency is required to increase security in the secret keys [47].

Another choice to infer the profile of the clients without utilizing app-level data, is modeling traffic data (payload) as a function of the users. Different black-box modeling techniques have been used with this purpose [12]. Almeida *et al.* [10] presented different simple models for describing the traffic of voice calls of a GSM network in Lisboa. Specifically, trapezoidal and Gaussian models are proposed to classify the areas as densely urban, urban and suburban. This work has been referenced by subsequent research works that have developed models of the behavior of 2G [14, 15], 3G [16–18], 4G [11, 19, 66] and hybrid [20–24] networks.

Afterwards, Pina [11] uses the approaches presented in [10] for constructing space-time models of data traffic in an LTE network, to characterize the consumption profile of the users based on the applications they employ. This information is useful to optimize the network in terms of quality of service (QoS). Note that in [10] and [11] a type of model is assumed a priori, and parameters that will make that model fit a particular data of traffic are further calculated. Although simple, this approach is rather restrictive due to limited number of models available [51].

Now, in mobile communication networks, traffic is significantly higher in most populated cities, where it follows a “swinging” behavior between residential areas, regularly located in the outskirts of the city, and working/commercial areas in the downtown. The traffic of data and calls also varies for the different locations and days of the week [51]. Therefore, the performance of the models discussed in [12, 25–39] may be affected by this nonlinear and time-varying dynamics.

On the contrary, neural networks-based models are a promising alternative for successfully solving this modeling task. First, neural network models can approximate any function with arbitrary degree of accuracy [67]. In addition, these approaches have exhibited excellent performance in analyzing, extracting information and identifying models from large amounts of data [51]. At last, time-varying features can be obtained through on-line training of the neural network model.

In fact, different types of neural networks have been applied to create models of traffic from measurements, as summarized in the survey by Zhang *et al.* [12]. Yazti and Krishnaswamy [13] present research, practice and opportunities of big data analytics applied to mobile communication networks. Some works ([14–19, 66]) describe models for optimizing the allocation of access channels and of the handover in the mobility and retain ability of the connection. Other works ([20–39]) propose predictive models that use the traffic of the cells, in the search for optimizing the distribution of RAN and Core Network resources based on QoE. Additional works present space-time models of traffic [40–46]. All these models are then further used to optimize load balance and handover [33–39] in the various technologies (2G, 3G, 4G, 5G and WiFi), and to predict the occurrence of undesired events such as network congestion in certain nodes ([25–39]).

Training speed is a key aspect when building a time-varying neural network model [68]. Even though standard backpropagation and its variants are the training algorithms most commonly used in the reported work [68], the speed issue is against these gradient-based algorithms that typically converge slowly because they require several iterations. In addition, their performance depends critically on training parameters that must be specified by the user, without well-established rules for selecting their values for a given training task [68].

This dissertation proposes the use of advanced time-varying approaches for modeling the dynamics of the multipath channel in wireless communication networks. First, the two initial case studies presented involve the use of linear time-varying Kalman innovation models for (i) describing measured RSSI in indoor multipath wireless channels [48] and (ii) secret key generation in the physical layer of multipath wireless channels [47]. On the other hand, the third case study evaluates the performance of two time-varying neural network-based modeling schemes for describing traffic in mobile telephony networks, namely a neural network-based autoregressive with exogenous inputs (NN-ARX) model [51], and a neural network-based autoregressive moving average with exogenous inputs (NN-ARMAX) model. The training speed issue in the neural network-models is resolved with the use of the online sequential extreme learning machine (OS-ELM) algorithm ([69, 70]).

With respect to the last case study, it will be shown that the time-varying NN-ARMAX model may yield a linear time-varying Kalman innovation model, as opposed to the time-varying NN-ARX model. It is important to remark that the NN-ARMAX model features the modeling power of neural networks, while the obtained equivalent Kalman innovation model enables the implementation of advanced schemes [68] for controlling the operation of the wireless networks.

1.2 Contribution

This dissertation proposes the use of advanced time-varying approaches for modeling the dynamics of the multipath channel in wireless communication networks. These advanced time-varying approaches include the linear Kalman innovation model in observable block companion form, and two neural network-based models, namely the neural network-based autoregressive moving average with exogenous inputs model and the neural network-based autoregressive with exogenous inputs model.

Previous published works that apply Kalman models for channel estimation ([1–4]) and key generation ([5,7–9,65]) in wireless networks use the classical Kalman filter. Compared to such classical Kalman approach, the Kalman innovation model used in this work has the advantage that it does not require to estimate the properties of the noise, while maintaining the typical estimation accuracy and time-varying features.

On the other hand, the neural network-based models used here outperform other black box approaches using simple models such as double-Gaussian [10] and trapezoidal [11]. In addition, compared to previous works that use neural networks approaches for modeling the dynamics of wireless networks ([12–46, 66]), the NN-ARX and NN-ARMAX models trained using OS-ELM exhibit an outstanding accuracy and significantly greater training speed; the latter makes them very suitable for identifying models for describing time-varying dynamics of different phenomena in wireless communication networks.

1.3 Summary of the Dissertation

The rest of the dissertation is structured as follows. Chapter II presents some theoretical fundamentals of the research work. Specifically, notions related to wireless communications are first depicted, including general concepts, important aspects of design, Global System for Mobile Communications (GSM), propagation properties (reflection, diffraction, dispersion and absorption), propagation in closed environments and multipath. Then, some considerations about secrecy and security in wireless systems are provided, including notions of secret key generation. At last, different schemes for modeling wireless communication systems from data are presented, ranging from simple models to describe the dynamics in time (double Gaussian and trapezoidal) and in space, to more advanced time-varying approaches such as linear Kalman innovation modeling and neural network-based modeling; these advanced approaches are the main subject of interest of this work.

Afterwards, Chapter III first describes the three case studies, which include (i) linear Kalman innovation modeling of indoor multipath wireless channels, (ii) secret key generation in the physical layer in multipath wireless channels and (iii) time-varying neural network-based modeling schemes for describing traffic in mobile telephony networks. Then, this chapter presents the criteria that will be utilized to evaluate the performance of the advanced modeling approaches.

At last, chapter IV presents and analyzes the results obtained in the different case studies, followed by concluding remarks that include further directions of research.

Chapter 2: SOME THEORETICAL FUNDAMENTALS

2.1 Wireless communications

This sub-section presents some basic concepts about wireless communication systems. The concepts presented are based on Goldsmith [71].

2.1.1 General concepts

Wireless communications are defined as any connection that enables the exchange of information between two or more points through radiation. In particular, by means of electromagnetic waves, with the air as the channel. Based on the location of the communicating sites they are classified as terrestrial or satellite, and according to their mobility, they are classified in fixed, mobile and nomadic.

This research considers mobile or fixed wireless terrestrial communications. Specifically, the case studies comprise, without loss of generality, WiFi access services, and GSM mobile telephony service, including the evolutionary technologies UMTS, HSPA, HSDPA and LTE.

In general, the transmission channel in wireless communications uses air as the medium, but there may be possibly obstacles in the trajectory. Propagation in free space refers to the case where there are virtually no obstacles, and the media with this geometry are called rural. On the other hand, when there is a complicated geometry between the communicating sites they are known as urban, which includes indoor propagation. The multipath phenomenon occurs in the urban case, due to the multiple trajectories that a transmitted signal follows in its way to the receiver because of its reflection on the multiple obstacles.

The maximum transmission speed of any communicating path is given by the capacity of the channel, which is defined as [50]

$$C[\text{bps}] = B \log_2\left(1 + \frac{S}{N}\right) \tag{2.1}$$

where C is the capacity of the channel in bits per second (bps), B is the bandwidth in Hertz (Hz) and S/N is the signal-to-noise ratio. Equation (2.1) clearly shows that the maximum capacity of the channel, i.e. its maximum allowed speed, is proportional to the signal-to-noise ratio, which becomes the parameter under study.

In general, it is necessary to model the communication channel to predict the behavior of the signals, for the design and control of the signal-to-noise ratio, among other applications. In the signal-to-noise ratio, the noise is a random component. In addition, in channels with multipath, the behavior is completely random; consequently, building models using physical principles is difficult, and data-driven models become a promising alternative. In other words, describing the channel in wireless communication processes with multipath may be more successful with the use of mathematical models identified from measured data.

2.1.2 Important aspects of the design

For the design of any wireless system, it is first necessary to define if it is mobile or fixed. Without loss of generality, in this work the GSM is considered as the case study for mobile systems, while WiFi is considered for fixed services. As a main feature, the wireless systems under consideration may be divided in two subsystems (with their particularities) known as core network (CN) and radio access network (RAN).

Obviously, the work is applied on the RAN, which is defined as the interconnection between the service provider and the final user of the wireless channel. Some characteristics of the GSM system will be described to illustrate basic concepts of wireless access networks.

2.1.3 Global System for Mobile Communications (GSM)

A GSM system, and in general any cellular telephony service establishes a communication between two mobile stations (cell phones), or between a mobile station and a fixed system.

As shown in Figure 2.1, a GSM system is a network further subdivided in two subsystems, namely the network subsystem (NSS) from the service provider to the interfaces of the base station controller, and the base station subsystem (BSS) from the base station controller to the user. The communication between the mobile terminal units and the base transmission station (BTS) is known as radio access network; in any wireless communication technology, the RAN is the communication between the users and the network. A full duplex communication, i.e. with uplink (UL) and downlink (DL) channels, is established in the RAN.

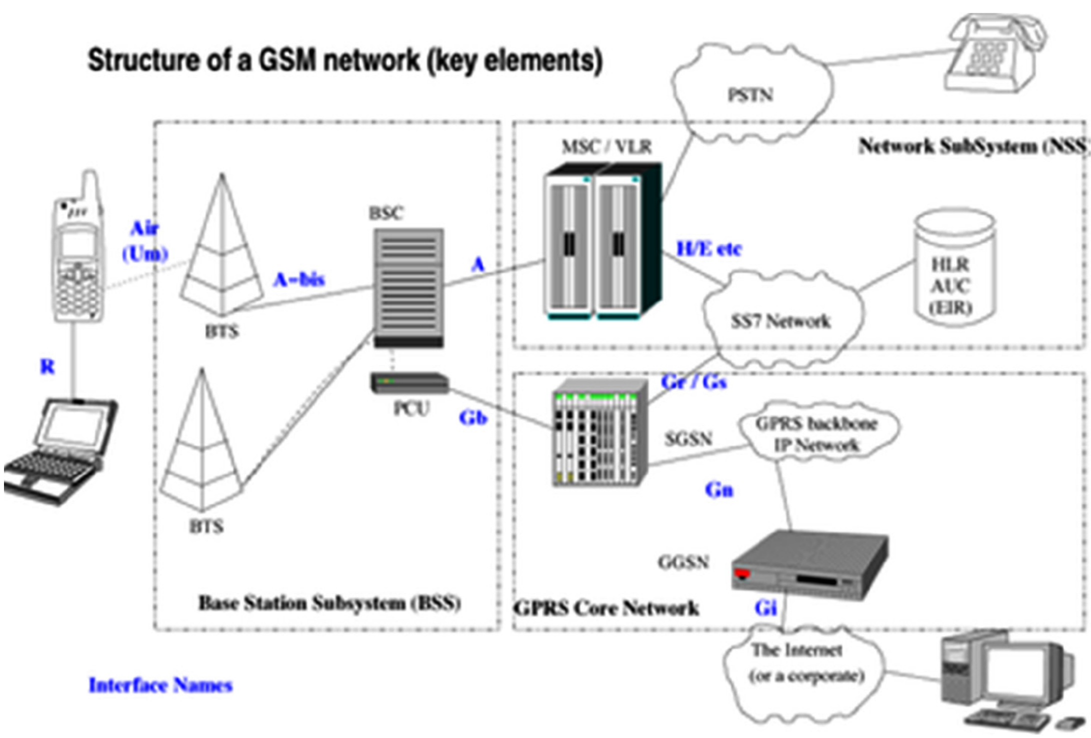


Figure 2.1: Basic architecture of a GSM system [49].

2.1.3.1 Signals

A signal is a wave that carries information. For the case of telecommunications, it refers to an electromagnetic wave that is distinguished from the others, because its variability is such that it describes a pattern that can be decoded by a receiver that transforms it to understandable and useful data.

In this work, signal refers to an electromagnetic wave that enables the communication between the access network and the final user in both directions, namely:

- Uplink (UL) channel: The transmitter is the final user and the receiver is the node (cell) in the radio access network.

- Downlink (DL) channel: The transmitter is the node (cell) in the radio access network and the receiver is the final user.

2.1.3.2 Coverage

For a wireless system, the coverage refers to the geographic region where there is available a service offer through the propagation of signals in the downlink (DL) channel. As will be shown below, the manner in which the coverage is measured depends on the technology of the network, i.e. 2G or GSM, 3G or UMTS, 4G or LTE, or WiFi Hotspot. In addition, in each technology the signals in the DL channel have different bandwidth and a different way to use such bandwidth to provide access to various users simultaneously.

The frequency hopping technology is employed in the case of GSM, which consists of using carriers of 200 KHz that transmit a portion of the information along an instantaneous time. Those carriers "hop" in a pseudorandom sequence in such a way that the never interfere each other. Besides, between them there exists a control carrier per cell called Broadcast Control Channel (BCCH) which is fixed, unique for every cell, and carries out all the call signaling and control.

In terms of engineering of mobile telecommunications, the coverage in GSM is measured as the level of power (in dBm) of the downlink signal of the strongest BCCH at each geographic point. Such level of power is known as Rx level, and its reading ranges are shown in Table 2.1.

Table 2.1: Rx Level (GSM)

Signal level in dBm	**Qualification of Coverage**
> -75	**Good Coverage**: It guarantees access and good quality
> -85 and < -75	**Regular Coverage**: It guarantees access, and guarantees quality in voice but not in data
> -95 and < -85	**Access Coverage**: It guarantees access, but does not guarantee quality in voice neither in data
< -95	**Poor Coverage**: It does not guarantee access

In the case of 3G or UMTS, the access technology is known as wideband code division multiple access (WCDMA), which utilizes all the bandwidth for all users at the same time. Access is provided to all users simultaneously, through different codes for each user and different codes for each cell. In this way, all signals are being transmitted at the same time, but they do not interfere each other because of the orthogonality between the codes of the different users, and the orthogonality between the codes of the different cells.

Table 2.2: RSCP (UMTS)

Signal level in dBm	**Qualification of Coverage**
> -75	**Good Coverage:** It guarantees access and good quality
$> -85y < -75$	**Regular Coverage:** It guarantees access, and guarantees quality in voice but not in data
$> -95y < -85$	**Access Coverage:** It guarantees access, but does not guarantee quality in voice neither in data
< -95	**Poor Coverage:** It does not guarantee access

The code corresponding to each cell is known as scrambling code, and each cell employs a portion of its power to provide each user with coverage and a specific control channel, called common pilot channel (CPICH). The level of the downlink signal of the CPICH is known as Received Signal Code Power (RSCP), which is the power received by the mobile phone after decoding the signal and is the parameter used to measure coverage in 3G. The reading ranges of the RSCP are shown in Table 2.2.

In the case of 4G or LTE, the access technology is known as Orthogonal Frequency Division Multiple Access (OFDMA). In this technology the whole bandwidth is utilized simultaneously for all the users, and the way to provide access to all the users at the same time is through different carriers of 200 kHz each, but orthogonal between them to avoid interference. Each cell is distinguished from the other based on its synchronization sequence, called physical cell indicator (PCI). The level of the downlink signal is known as Radio Strength Resource Power (RSRP), which is measured in dBm, and its reading ranges are shown in Table 2.3.

Table 2.3: RSRP (LTE)

Signal level in dBm	Qualification of Coverage
> -85	**Good Coverage**: It guarantees access and good quality
$> -95y < -85$	**Regular Coverage**: It guarantees access, and guarantees middle quality in data
$> -100y < -95$	**Access Coverage**: It guarantees access, but does not guarantee quality in data
< -100	**Poor Coverage**: It does not guarantee access

In the 3G and 4G technologies, the system operates broadcasting signals according to the traffic. If the traffic begins to grow so much that the quality of service might be degraded, the network reduces the power of the RSCP (3G) or RSRP (4G) to decrease the quantity of users, and retaining only the number of users to which a good quality of service can be guaranteed. In contrast, GSM operates with a constant Rx Level. Regarding the WiFi technology, it also uses OFDM for multichanneling the Downlink band, and thus it exhibits certain behavior similar to the LTE.

Based on the previous considerations, it can be noted that the received signal strength indicator (RSSI) is the variable common to all technologies to measure coverage. The RSSI is defined as the power of the signal (in dBm) right before entering the receiver, and it can be measured using a spectrum analyzer, a sniffer or any smart mobile device with an appropriate application to store the measurements.

2.1.4 Propagation properties

The propagation of electromagnetic waves is the most used medium for transmitting information in telecommunication systems. Some phenomena related to propagation in mobile telephony systems are now pointed out.

Most communication systems use very complex propagation mechanisms, and it is difficult to find a model that describes such mechanisms in a precise manner. The basic propagation mechanisms present in wireless communication systems are reflection, diffraction, dispersion and absorption. These mechanisms are now briefly described.

2.1.4.1 Reflection

Reflection occurs when an electromagnetic wave that propagates through air hits an object of large dimensions, compared to the wavelength of the signal. This response depends mainly on: (i) physical properties of the object, such as geometry, texture and composition, and (ii) properties of the signal, such as angle if incidence, orientation and wavelength.

Perfect conductors will entirely reflect the signal. Other materials reflect a part of the incident energy and transmit the rest. The exact amount of transmission and reflection depends of the angle of incidence, the thickness and dielectric properties.

When an electromagnetic signal is transmitted through air, most probably it will reach the receiver following multiple paths. The signals coming from such alternative paths will arrive slightly delayed and with smaller amplitude with respect to the direct signal; this causes a fading effect.

2.1.4.2 Diffraction

Diffracted waves are formed when the propagation path of the radio wave is obstructed by an impenetrable object whose surface is irregular, or has sharp or angled edges. Based on the principle of Huygens, the result are secondary waves around and behind the obstacle, even in zones without direct visibility between transmitter and receiver.

Closed environments contain many types of objects with these characteristics, oriented both in the vertical and horizontal planes. The diffracted signal depends on the geometry of the object, and on the amplitude, phase and polarization of the incident wave at the point of diffraction.

The Fresnel zones represent successive regions where the path of the secondary waves from the transmitter to the receiver has a length $n\lambda/2$ larger than the total length of the direct path, where n represents the layer of the Fresnel zone and λ is the wavelength. The Fresnel zones explain the concept of diffraction losses as a function of the distance of the path around the object.

In wireless systems the diffraction losses are due to the obstruction of secondary waves, such that only a portion of the energy is diffracted around an obstacle. In other words, an obstruction causes a blockage of energy from some of the Fresnel zones, and consequently only part of the transmitted energy reaches the receiver. Depending on the geometry of the obstruction, the received energy will be the vector sum of the energy contributions of all Fresnel zones not being obstructed. In general, if 55% of the first Fresnel zone is kept clear, then the zone of free space beyond such Fresnel zone does not significantly alter the diffraction losses.

2.1.4.3 Dispersion

Dispersion occurs when the signal finds in its path objects whose dimensions are small compared to the wavelength. As a result, the wave front breaks or disperses in multiple directions.

The disperse waves are produced by uneven surfaces, small objects and other irregularities in the channel. The structure of most modern constructions contain forged iron beams, water pipes and ducts for electrical services. In practice, foliage, traffic signs or lanterns may produce dispersion in wireless communication systems.

2.1.4.4 Absorption

Absorption occurs when part of the electromagnetic energy becomes heat, during propagation of the wave. This is a consequence of the polarization due to the orientation of the water molecules, which appears for frequencies in the microwave and radio waves bands. Other types of polarization, namely ionic and electronic, are produced at other frequencies (infrared and ultraviolet).

2.1.4.5 Losses due to penetration in a closed environment

One of the most precise concepts of these losses defines them as the difference between the average of the signal measured at the ground of a building, and the average of the signal measured at the floor of interest.

The propagation in indoor communications systems is influenced by parameters associated to the construction, such as thickness of the walls, materials and internal structures; these parameters and the diffraction coefficients of many internal structures are seldom known. This lack of knowledge also hampers the possibility of using simulation tools to get precise, secure and computationally efficient predictions, for the purpose of devising a strategy to reduce the indoor RF propagation losses.

In addition, the propagated signal frequently finds many obstacles, thus reflecting and generating multiple paths to the receiver. As will be described below, this phenomenon is random and time varying, thus making difficult modeling the dynamics of the channels. Besides, attempts to trace the signal solely based on propagation models may be inadequate.

Based on the above, analyzing and forecasting propagation features in a closed environment poses challenges. As a consequence, this topic is of great interest for researchers in the area of radiofrequency (RF) propagation.

2.1.4.6 Propagation in a closed environment

The propagation of electromagnetic waves may occur both outdoor or in closed environments. The former is influenced by atmospheric conditions such as clouds, rain and snow, among others. On the other hand, in closed environments it is mainly affected by the construction materials and by the geometric configuration of the space.

When studying the propagation in a closed environment, the configuration of both the sites where the communication takes place and of the coverage zones where the network services are provided, with the purpose of physically characterizing the environment (office, home, among others).

Phenomena such as reflection, diffraction, dispersion and absorption in the transmitted wave, because of obstacles in the path of the signal, should be considered in the case of propagation in closed environments. This results in the signal reaching the receiver through more than one path, in which there may or may not be direct line of sight from the transmitter to the receiver, even if such signals travel short distances. The former situation will make more difficult determining parameters such as the capacity of the channels or the quality of the communication links.

2.2 Multipath

2.2.1 Overview

Multipath is a phenomenon that appears when the electromagnetic waves propagate in environments with complex geometry, in which multiple reflections generate multiple paths between transmitter and receiver. Therefore, the information reaches the receiver various times at different instants, following a random pattern. Therefore, deterministic channel models are rarely available in practice, thus multipath channels must be characterized statistically [49], and it is necessary to model them by means of a random time-varying impulse response [72].

Multipath channels are often modeled as a linear time-invariant channel over a limited time interval. Even though diffuse scattering can occur and would be modeled by a continuous impulse response, in most cases the channel is modeled as tapped delay line with impulse response given by

$$h(\tau) = \alpha_0 e^{-j\theta_0}\delta(\tau - \tau_0) + \sum_{n=1}^{N} \alpha_n e^{-j\theta_n}\delta(\tau - \tau_n) \tag{2.2}$$

where α_0, θ_0 and τ_0 are the power, carrier phase and propagation delay, respectively, of the signal received in the direct path, while α_n, θ_n and τ_n $(n = 1, \ldots, N)$ are the power, carrier phase and propagation delay, respectively, of the N signals received through other paths; besides, each of these additional paths have associated a Doppler frequency. Note that the sum of impulses from the different paths depends on random elements such as the delay, amplitude and phase of those paths. Depending on the random phase shift associated with each received signal, they might add up destructively, resulting in a phenomenon called fading [6].

Then, the received signal $y(t)$ can be represented in compact form as

$$y(t) = \sum_{i=1}^{N} \alpha_i s\big(t - \tau_i(t)\big) \tag{2.3}$$

where N is the number of rays arriving to the receiver, $s(t)$ is the bandpass input signal, α_i is the path attenuation and τ_i is the path delay. If $s(t)$ is written as

$$s(t) = \mathrm{Re}\left\{\widetilde{s(t)}e^{j2\pi f_c t}\right\} \tag{2.4}$$

the complex channel output is given as

$$\tilde{y}(t) = \sum_{i=1}^{N} \widetilde{\alpha}_i \tilde{s}\big(t - \tau_i(t)\big) \tag{2.5}$$

where $\widetilde{\alpha}_i = \alpha_i e^{j2\pi f_c t}$, with f_c the frequency of the carrier.

Therefore, the time-varying discrete multipath channel can be described by the time varying complex impulse response:

$$\tilde{h}(\tau,t) = \sum_{i=1}^{N} \widetilde{\alpha}_i \delta\big(t - \tau_i(t)\big) \tag{2.6}$$

where $\widetilde{\alpha}_i$ is the time varying complex attenuation of each path. It can be seen that given a fixed number N of paths and path delays τ_i, based on equation (2.6) it is possible to characterize the time-varying channel if the properties of the complex attenuation $\widetilde{\alpha}_i$ for each path are specified.

Based on the assumption of wide sense stationary uncorrelated scattering (WSSUS), the delayed cross power spectral density is defined as:

$$R_c(\tau,\Delta t) = \frac{1}{2}E\left[\tilde{h}^*(\tau,t)\tilde{h}(\tau,t+\Delta t)\right]. \tag{2.7}$$

The average power as a function of path delay can be obtained considering $\Delta t = 0$ in (2.7), i.e.

$$R_c(\tau,\Delta t) = R_c(\tau). \tag{2.8}$$

The function $R_c(\tau)$ is known as Multipath Intensity Profile or Delayed Power Spectrum. The range of τ for which $R_c(\tau)$ is essentially non-zero is called the Multipath Delay Spread. Another important characterization of the channel is the Scattering function, which describes the relationship between the power spectrum and the path delays, and is represented as

$$S(\tau,v) = F\left\{R_c(\tau,\Delta t)\right\} = \int_{-\infty}^{\infty} R_c(\tau,\Delta t)e^{-j2\pi v\Delta t}d(\Delta t) \tag{2.9}$$

where $F\{x\}$ is the Fourier transform operator. For fixed τ, the scattering function describes the power spectral density in the frequency variable v, referred to as the Doppler Frequency.

Considering again the tapped delay line model (2.2), if it is assumed that the number of scatterers in each path is infinitely large, according to the central limit theorem (as will be seen later) $\tilde{h}(\tau,t)$ can be modelled as a complex Gaussian process. If this process is assumed to be zero mean, then the envelope $|\tilde{h}(\tau,t)|$ follows a Rayleigh distribution, and the channel is said to be a Rayleigh fading channel. If a part of the signal reaches the receiver directly and the remaining arrives to it through a continuum of paths, then $\tilde{h}(\tau,t)$ can be modeled as a Gaussian process with nonzero mean, which corresponds to a ricean-fading channel. Therefore, in order to represent the impulse response of a multipath radio channel, the tap gains have to be chosen as sampled versions of a complex Gaussian process. Then, the tapped delay model has the following characteristics [6]:

- The number of taps is $T_M W + 1$, where T_M is the delay spread and W is the information bandwith.
- The tap spacing is $1/W$, which is the resolution of the multipath channel model.
- The tap gain function $\tilde{g}(t)$ is a discrete time complex Gaussian processes with the variance of each component given by

$$\sigma_m^2 = \frac{1}{W^2} R_c \left(\frac{m}{W} \right) \tag{2.10}$$

- The PSD of $\tilde{g}(t)$ is

$$S_{gg} = \frac{1}{W^2} S\left(\frac{m}{W}, v\right) \tag{2.11}$$

In summary, the multipath channel is a discrete-time complex Gaussian process, and it should be estimated using techniques associated to stochastic processes applied to the data.

2.2.2 Description of the dynamics of the multipath Channel [47]

As it was described before, signal multipath occurs when the transmitted signal arrives at the receiver via multiple propagation paths, direct and (possibly) various reflection paths. Each path may have a different phase, attenuation, delay and Doppler frequency. Therefore, because of the randomness introduced by multipath, the channel between any pair of communicating users, Alice (transmitter) and Bob (receiver) is unique.

This random channel is typically described as a fading channel of Rayleigh type with Jake's model [64], with auto-correlation of the gain given by [6, 58]

$$Rgg(\tau) = E\Big[g(t)g(t+\tau)\Big] = J_0(2\pi f_d \tau) \tag{2.12}$$

where $J_0(\cdot)$ denotes the zeroth order Bessel function of the first kind, and f_d is the maximum

Doppler frequency. The power spectral density (PSD) of this fading process is [6, 58]

$$P_{gg}(f) = \frac{1}{\pi f_d \sqrt{1-(f/f_d)^2}} \quad \text{for} \quad |f| < f_d. \tag{2.13}$$

2.3 Security in Wireless Communication Systems

2.3.1 Overview

The theory of secure communication systems is based on various works in the area of Information Theory. Some basic assumptions related to secure coding are: (i) a third element, known as Eavesdropper (Eve), tries to obtain the message, (ii) the channel is vulnerable to Eve and (iii) Eve knows how to demodulate the information.

Figure 2.2 shows the schematic of a general secrecy system. Note that the Encipherer has two sources, message source and key source. The message source is the origin of the communication, while the key source is the coder of a sufficiently random key, to prevent it from being discovered. The Encipherer takes both signals and mixes them in a pseudorandom manner, thus generating a signal known as cryptogram, which is finally carried to the channel through a transmitter.

The Eavesdropper, who is in the channel, is considered an Enemy Cryptanalyst, i.e. a receiver with the ability of deciphering and demodulating the transmitted signal. However,

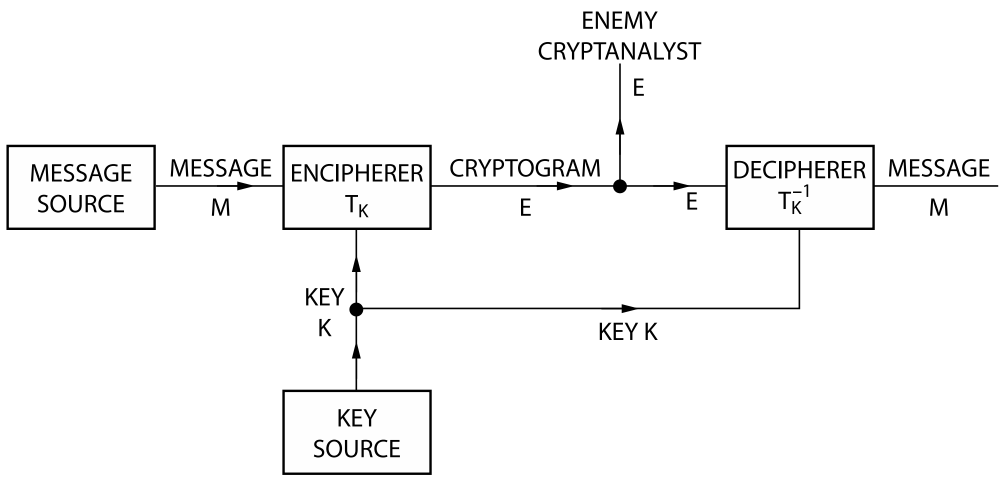


Figure 2.2: Schematic of a general secrecy system [50]

that should not be possible because of the randomness of the Cryptogram.

In the receiver, the Decipherer receives the transmitted signal and demodulates and deciphers it; this can be done because the key has been received through other secure communication channel. Then, the original message is obtained from the message source.

The Key Source should have an entropy sufficiently large to mislead Eve at all times, and the key should be continuously updated to prevent the Eavesdropper from learning it. The entropy is a measure of the randomness of a set of symbols. Now, two definitions associated to this process are stated.

Definition 1. *A length-N secret key generation system over alphabets X_A, X_B, K, S is a triplet of functions: (i) $f : X_A^N \to K^{NB}$, which maps Alice's source of randomness into the secret key, (ii) $g_A : X_A^N \to S^m$, which defines the public message that Alice sends to Bob and* *(iii)* $g_B : X_B^N S^m \rightarrow X_A^N$*, which is Bob's decoding function that maps his observation and the public message into his estimate of Alice's observation. If Bob's estimate is correct, the key will be recovered [59].*

Given a source of randomness $p_{X_A^N, X_B^N}(x_A^N, x_B^N)$, where $x_A^N \in X_A^N$ and $x_B^N \in X_B^N$, the secret key capacity is the supremum of the achievable secret key rates. An achievable secret key rate is defined as follows.

Definition 2. *For any sufficiently large secret key, the key rate is achievable if the following inequalities hold:*

i $NR \log |K| - H(f(X_A^N)) \leq \varepsilon$*, which implies that the secret key is nearly uniformly distributed*

ii $\Pr[f(X_A^N) \neq f(g_B(X_B^N, g_A(X_A^N)))] \leq \varepsilon$*, which represents an upper bound for the probability error in key recovery.*

iii $\frac{1}{N} I(f(X_A^N), g_A(X_A^N)) \leq \varepsilon$*, which is the secrecy guarantee, i.e., it guarantees that the public message tells little about the key.*

2.3.2 Secret key generation in the physical layer

The algorithm for generating secret keys through the physical layer in wireless systems comprises six steps [47]:

Step 1. Initialization or beacon exchange. Both Alice and Bob start to exchange the signal that will be used to estimate the physical layer characteristic. Multiple exchanges might be necessary based on the required length and rate of the key.

Step 2. Estimation of the common source of randomness. Based on the received signal from the other legitimate node, Alice and Bob estimate the variable that constitute the common source of randomness, using a model of the channel. The variable that will be used for key generation should be the same for Alice and Bob, and both must generate as similar as possible measurements of the variable (with slight deviations due to different noise at the measuring sites).

Step 3. Quantization. Alice and Bob map the estimated value of the variable into one of 2^n possible discrete levels, where n is the number of bits used to encode the secret key. The most popular technique for quantization is the uniform quantization, which defines equally spaced discrete levels.

Step 4 Encoding. Alice and Bob convert the quantized value into a string of bits, which correspond to the secret key. The conventional secret key length is between 128 and 512 bits.

Step 5. Information Reconciliation. Consists of using protocols, which involve permutation and parity check algorithms, to increase security by minimizing the discrepancy between the bit streams generated by Alice and Bob (particularly at very low SNR levels).

Step 6. Privacy Amplification. The eavesdropper can still use minimum leaked information from reconciliation, to guess the rest of the secret key. In this step Alice and Bob use the same randomly selected hash function to solve this issue, generating a shorter but higher in entropy bit stream.

Alice and Bob cannot take simultaneous measurements, since standard radio transceivers cannot simultaneously transmit and receive. Denote as M_u the number of measurements of the channel gains that each user makes before switching modes, and select the block length M to be an integer multiple of $2M_u$. Denote the indices of Alice's measurements as

$$n^A = \left[1 \ldots M_u, 2M_u + 1 \ldots 3M_u, M - 2M_u + 1 \ldots M - M_u\right]^T,$$

and the indices of Bob's measurements as $n^B = [M_u + 1 \ldots 2M_u, 3M_u + 1 \ldots 4M_u, M - M_u + 1 \ldots M]^T$ [58].

At the corresponding indices, each radio transceiver will get noisy measurements of the channel RSSI described by

$$\hat{g}^X(n^X T) = g(n^X T) + e^X(n^X T) \tag{2.14}$$

where the superscript X is A for Alice or B for Bob, $g(n^X T)$ is the true RSSI measured value, $e^X(n^X T)$ is zero mean additive white Gaussian noise (AWGN) of variance σ^2 and independent of the channel gain process, and T is the sampling period [58].

2.4 Wireless Communication Systems modeling from data

2.4.1 Simple space-time traffic models

Time and space traffic models are being studied from the beginning of the wireless communications through GSM.

2.4.1.1 Time traffic models

Initial works were carried out to investigate functions that exhibit a good fit the dynamics of the demand in time. The double-Gaussian ([10, 11]) and trapezoidal functions ([6, 10, 11, 43]) were used to obtain models of the traffic as a function of time. The double-Gaussian function [10] is a function of generalized use, and consists of two Gaussian functions centered at the morning and afternoon busy hours, respectively, with a breakpoint at the lunch hour.

The double-Gaussian function, which is shown in Figure 2.3, is defined as

$$ap_{\text{Gauss}}(t_{sh}) = \begin{cases} p1 \cdot e^{-\frac{(t_{sh}-h_{1sh})^2}{2d_1^2}}, & \text{for} \quad t_{sh} < h_{lsh} \\ p2 \cdot e^{-\frac{(t_{sh}-h_{2sh})^2}{2d_2^2}}, & \text{for} \quad t_{sh} > h_{lsh} \end{cases} \tag{2.15}$$

where $p1$ is the amplitude of the first Gaussian, h_{1sh} is the shifted morning peak hour, d_1^2 is the variance of the first Gaussian, h_{lsh} is the shifted lunch hour, $p2$ is the amplitude of the second Gaussian, h_{2sh} is the shifted afternoon peak hour and d_2^2 is the variance of the second Gaussian. Continuity is ensured at the transition point, where ap takes the minimum value of the two Gaussians.

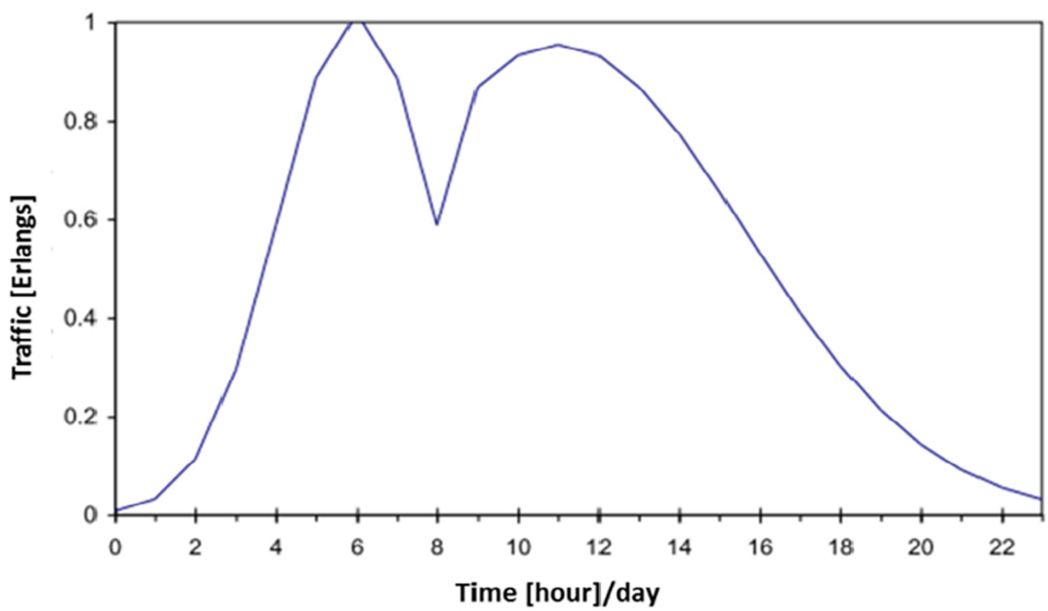


Figure 2.3: Double-Gaussian model [11].

The trapezoidal function is an upper limited Gaussian, which is more appropriate in areas were the evening traffic has a significant weight [10]. This function is shown in Figure 2.4 and is defined as

$$ap_{\text{Trap}}(t_{sh}) = \begin{cases} p \cdot e^{-\frac{(t_{sh}-h_{tsh})^2}{2d_t^2}}, & \text{for} \quad t_{sh} < t_{q1sh} \\ c, & \text{for} \quad t_{q1sh} \leq t_{sh} \leq t_{q2sh} \\ p \cdot e^{-\frac{(t_{sh}-h_{tsh})^2}{2d_t^2}}, & \text{for} \quad t_{sh} > t_{q2sh} \end{cases} \tag{2.16}$$

where p is the amplitude of the Gaussian, h_{tsh} is the shifted Gaussian peak hour, d_t^2 is the variance of the Gaussian, c is the upper limit, t_{q1sh} is the shifted first breakpoint and t_{q2sh} is the shifted second breakpoint. Continuity is ensured at both breakpoints [10].

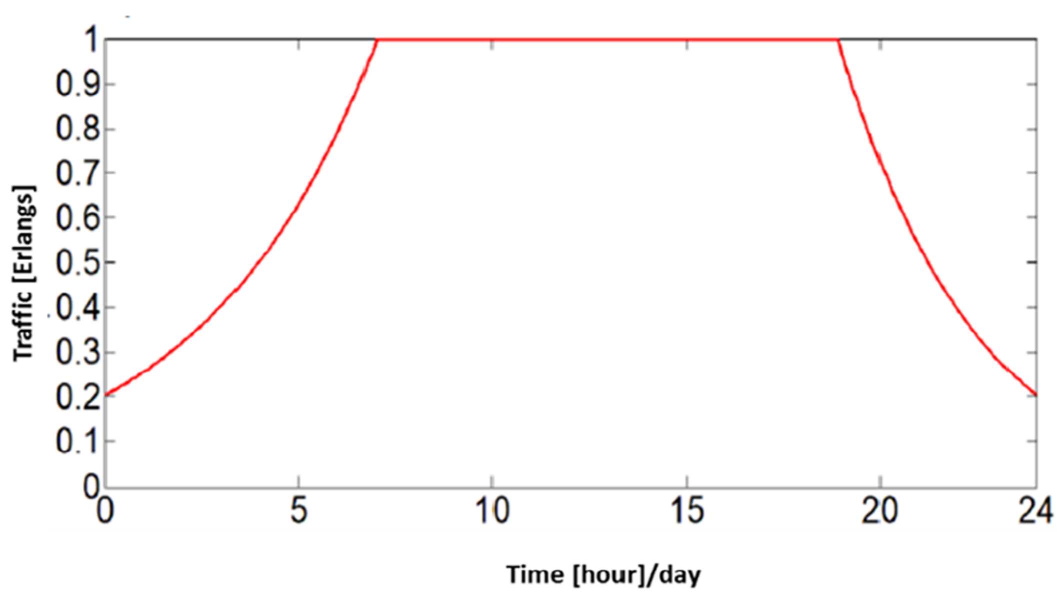


Figure 2.4: Trapezoidal model [11].

2.4.1.2 Space traffic models

This type of models consist of a configuration of the geographic distribution of stations and users in the area of coverage. They are obtained according to the following procedure.

First, geographic quadrants are generated in which it is assumed that the superficial density of the traffic is constant. The dimension of such quadrants is defined based on the population density of the area of interest; a dimension of $50\,\mathrm{m} \times 50\,\mathrm{m}$ is typical. Then, the quadrant with the largest concentration is located and, at last, it is studied the traffic density in different directions from this point of largest concentration. These models are updated every hour, to describe the spatial behavior of the traffic in a static manner. A dynamic modeling requires a more thorough study.

To describe the variation of the traffic density as a function of distance, three models have been tested, namely, exponential, given by

$$\mathrm{mod}_{\,e}(d) = e^{-d/D_e} \tag{2.17a}$$

exponential/linear, defined as

$$\mathrm{mod}_{\,el}(d) = \begin{cases} e^{-d/D_{el}}, & d \leq dq \\ \mathrm{Cel}, & d > dq \end{cases} \tag{2.17b}$$

and piecewise linear, given as

$$\operatorname{mod}_{pl}(d) = \begin{cases} 1 - Apl \cdot d, & d \leq dq_1 \\ Bpl - Cpl \cdot d, & dq_1 < d \leq dq_2 \\ Dpl, & d > dq_2 \end{cases} . \tag{2.17c}$$

In (2.17), D_e is the exponential decay factor, D_{el} is the exponential/linear decay factor, dq is the exponential/linear breakpoint, Cel is the exponential/linear constant factor, Apl is the piecewise first piece slope, dq_1 is the piecewise first breakpoint, Bpl is the piecewise second piece constant, Cpl is the piecewise second piece slope, dq_2 is the piecewise second breakpoint and Dpl is the piecewise constant factor.

Even though these models were extensively used during the expansion of GSM at the beginning of 2000s, because they exhibited the best fit at the time, researchers concluded that they were inaccurate, and that it was necessary to develop more complex models that could yield a better representation of the traffic dynamics [10].

2.4.2 Advanced approaches for modeling traffic

2.4.2.1 Linear time-varying Kalman innovation model in observable block companion form

Canelon *et al.* [68] presented a linear Kalman innovation model in observable block companion form which is described next [47]. As opposed to the classical Kalman filter, the

Kalman innovation model used in this work has the advantage that it does not require to estimate the properties of the noise, while maintaining the typical estimation accuracy and time-varying features.

After the measurement $y(k+1)$ of the output at time instant $k+1$ is obtained, the aim is to identify and continuously update a linear autoregressive moving average with exogenous inputs (ARMAX) model of the form

$$\begin{aligned} \hat{y}(k+1) = -a_1^k y(k) - \cdots - a_p^k y(k-p+1) \\ +b_1^k u(k) + \cdots + b_p^k u(k-p+1) + d_1^k e(k) + \cdots + d_p^k e(k-p+1) \end{aligned} \tag{2.18}$$

where $u(k)$ is the measured value of the input and $e(k)$ is the innovation error, both at time instant k, and p is the number of previous values of the output, input and innovation error. Specifically, the vector of parameters

$$\theta_k = \left[\begin{array}{ccccccc} a_1^k & \cdots & a_p^k b_1^k & \cdots & b_p^k d_1^k & \cdots & d_p^k \end{array} \right]$$

of the linear ARMAX model is recursively updated using the extended least squares algorithm [68], i.e. by means of the formula

$$\theta_{k+1} = \theta_k + \frac{P_k \varphi_{k+1}}{\lambda_{k+1} + (\varphi_{k+1})^T P_k \varphi_{k+1}} e_{k+1} \tag{2.19}$$

where

$$\varphi_{k+1} = \Big[y(k) \quad \cdots \quad y(k-p+1)u(k) \quad \cdots \quad u(k-p+1)e(k)$$
$$\cdots \quad e(k-p+1)\Big]$$

is the data vector,

$$e_{k+1} = y(k+1) - (\theta_k)^T \varphi_{k+1} \tag{2.20}$$

is the innovation (estimation) error of the previous model on the actual measured value, P_k is a matrix updated using the formula

$$\mathbf{P}_{k+1} = \frac{1}{\lambda_{k+1}} \left(\mathbf{P}_k - \frac{\mathbf{P}_k \varphi_{k+1} (\varphi_{k+1})^T \mathbf{P}_k}{\lambda_{k+1} + (\varphi_{k+1})^T \mathbf{P}_k \varphi_{k+1}} \right) \tag{2.21}$$

and

$$\lambda_{k+1} = \lambda_a \lambda_k + (1 - \lambda_a) \tag{2.22}$$

is a forgetting factor with an initial value $0.9 < \lambda_0 < 1$, and an update factor $0 < \lambda_a < 1$.

Then, the time-varying linear Kalman innovation model in observable block companion form is given by [68]

$$\mathbf{x}_o(k+1) = \mathbf{A}_o^k \mathbf{x}_o(k) + \mathbf{B}_o^k u(k) + \mathbf{K}_o^k e(k) \tag{2.23a}$$

$$\hat{y}(k) = \mathbf{C}_o \mathbf{x}_o(k) \tag{2.23b}$$

where $\mathbf{x}_o(k) \in \mathbb{R}^{p\times 1}$, $u(k) \in \mathbb{R}$, $e(k) \in \mathbb{R}$ and $\hat{g}(k) \in \mathbb{R}$ are the observed state vector, the input, the innovation vector and the estimated output, respectively, at time instant k,

$$\mathbf{A}_o^k = \begin{bmatrix} -a_1^k & 1 & 0 & \cdots & 0 \\ -a_2^k & 0 & 1 & & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_P^k & 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{p\times p},$$

$$\mathbf{B}_o^k = \begin{bmatrix} b_1^k \\ b_2^k \\ \vdots \\ b_p^k \end{bmatrix} \in \mathbb{R}^{p\times 1},$$

$$\mathbf{K}_o^k = \begin{bmatrix} d_1^k - a_1^k \\ d_2^k - a_2^k \\ \vdots \\ d_p^k - a_p^k \end{bmatrix} \in \mathbb{R}^{p\times 1}$$

and $\mathbf{C}_o$ is fixed given by

$$\mathbf{C}_o = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \mathbb{R}^{1\times p}.$$

This procedure is repeated for every measurement.

The use of a forgetting factor will yield better performance for time-varying systems, due to the fact that the forgetting factor assigns smaller weights as the data is older. In addition, extended least squares is computationally more efficient since the update of $\mathbf{P}_k$

requires no matrix inversion [68].

2.4.2.2 Neural network-based models of dynamic systems

2.4.2.2.1 Definition and elements of a neural network

The development of artificial neural networks started in the 1940s, due to attempts of different research groups to model the behavior of biological nervous systems as highly interconnected systems of a large amount of simple elements that try to emulate the neurons. It was expected that a complex phenomenon such as intelligence emerged as a result of a training process. Nowadays, neural networks are included in a discipline known as Machine Learning.

Different definitions of artificial neural networks, or simply neural networks, have been presented in the literature. For instance, Fausset [73] states that a neural network (NN) is a system for information that has certain characteristics common with the biological neural networks. On the other hand, for Hecht-Nielsen [74], a NN is a structure for parallel and distributed information processing, constituted by elements known as neurons, units or nodes, which may have local memory and are interconnected through unidirectional connections. In addition, Haykin [75] defines a NN as a parallel, massive and distributed processing system, constituted by simple units that have the capability of storing knowledge obtained through experience; this knowledge may be used later. Specifically, such knowledge is stored in synaptic weights, or simply weights, which express the strength of connection between neurons. A training procedure modifies the weights in ordered manner, in order to fulfill the desired objective.

Neural networks have been successfully used to identify models of processes with complex dynamics; in general, this complexity is related to features such as nonlinearity, large number of variables, high dimension, uncertainty, among others, or it may simply be impractical to determine models based on physical principles. Indeed, neural networks are capable of approximating any functional relationship between variables, with arbitrary degree of accuracy [67].

A neural network is completely defined when the following elements are indicated:

a) Architecture: specifies the number of neurons and how they are connected to accomplish a particular task, and thus determines how the information flows through the network. Different architectures have been developed, including feedforward, radial basis functions, support vector machines and recurrent neural networks, among others. Feedforward neural networks are described with more detail in this work, since it has been the architecture with the greater amount of reported applications in modeling of dynamic systems, and are used here to describe the dynamics of wireless communication networks.

b) Transfer function: specifies how the output values of the network are calculated as a function of the inputs.

c) Training or learning algorithm: specifies how the weights are adjusted during the training process of the network. The learning algorithms or rules may be (i) supervised, in which the target values of the outputs are given, or (ii) unsupervised, in which those target values are unknown, and the network seeks to group the data in clusters, or extract probabilistic features.

2.4.2.2.2 Feedforward neural network (FFNN)

As previously mentioned, this is the model with the greatest number of applications related to the modeling of dynamic systems.

a) Architecture of the FFNN

It is constituted by various layers of nodes, connected sequentially. The FFNN comprises three types of layers: (i) one input layer, in which the input to the network is applied, (ii) one output layer, in which the output of the network is obtained and (iii) one or more hidden layers, which are located between the input and output layers. The flow of information is unidirectional, and occurs from input to output layer. According to a theorem by Kolmogorov [76], one hidden layer is enough to achieve arbitrary precision accuracy;

however, a hidden layer with many units may be replaced by various hidden layers of smaller size for practical purposes.

Figure 2.5 shows a FFNN with an input layer of I units (input vector of size I), a hidden layer of J units and an output layer of O units (output vector of size O). This FFNN is denoted as $I \times J \times O$. Each node of a layer is connected to all nodes of the subsequent layer, and each connection has associated a weight (represented as black squares in the figure) which is adjusted during the training process. The bias of each hidden and output unit is included as an additional weight whose input is always 1 (connections shown by discontinuous lines in the figure).

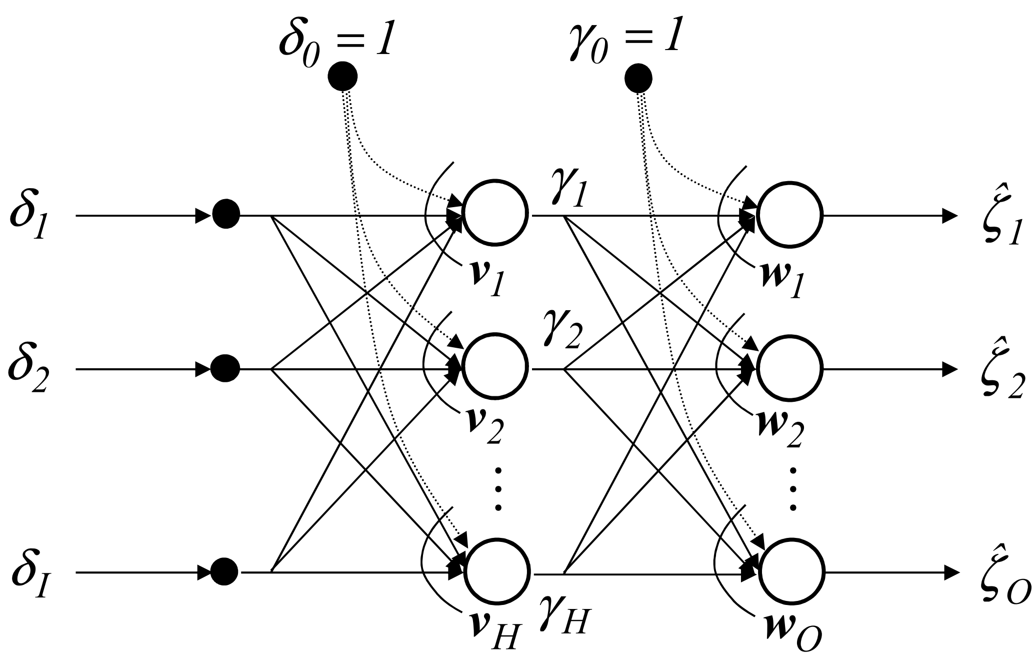


Figure 2.5: Feedforward neural network.

The weights of the connections between the input and hidden nodes are identified using letter v, and the weights between the hidden and output nodes are identified using letter w. In particular, v_{ji} represents the weight from the ith input unit to the jth hidden unit, while w_{oj} denotes the weight from the jth hidden unit to the oth input unit. Then, all weights between the input and hidden layers may be grouped in matrix

$$\mathbf{V} = \begin{bmatrix} v_{10} & v_{11} & \cdots & v_{1I} \\ v_{20} & v_{21} & \cdots & v_{2I} \\ \vdots & \vdots & \ddots & \vdots \\ v_{J0} & v_{J1} & \cdots & v_{JI} \end{bmatrix} \in \mathbb{R}^{J\times(I+1)} \tag{2.24a}$$

where vector $\mathbf{v}_j$, which corresponds to the jth row of $\mathbf{V}$, contains all weights directed to the jth hidden unit. Similarly, all weights between the hidden and output layers may be gathered in matrix

$$\mathbf{W} = \begin{bmatrix} w_{10} & w_{11} & \cdots & w_{1J} \\ w_{20} & w_{21} & \cdots & w_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ w_{O0} & w_{O1} & \cdots & w_{OJ} \end{bmatrix} \in \mathbb{R}^{O\times(J+1)} \tag{2.24b}$$

where vector $\mathbf{w}_o$ corresponds to the oth row of $\mathbf{W}$ and contains all weights directed to the oth output unit.

b) Transfer function of the FFNN

The input units do not carry out any processing, they only receive the inputs and distribute it to the units in the hidden layer.

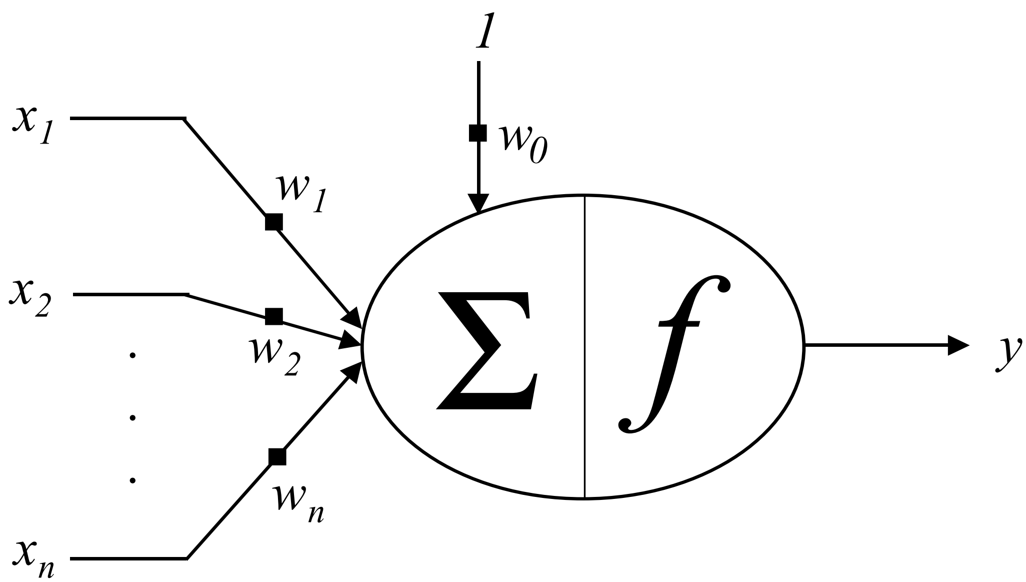


Figure 2.6: Hidden and output unit of a FFNN.

On the other hand, the unit shown in Fig. 2.6 is a general representation of the hidden and output units. The output y of the unit in the figure is calculated as

$$y = f(w_0 + x_1 w_1 + x_2 w_2 + \cdots + x_n w_n) \tag{2.25}$$

where x_i $(i = 1, \ldots, n)$ are the inputs, w_i $(i = 1, \ldots, n)$ are the corresponding weights, w_0 is the bias of the unit and f is the activation function of the unit. Therefore, if the input vector $\mathbf{x} = [1 \quad x_1 \quad x_2 \cdots x_n]^T$ and the weight vector $\mathbf{w} = [w_0 \quad w_1 \quad w_2 \cdots w_n]^T$ are defined, equation (2.25) may be rewritten as

$$y = f(\mathbf{x}^T \mathbf{w}). \tag{2.26}$$

The activation functions of the output units may be chosen as the sigmoid (shown in Figure 2.7a), the hyperbolic tangent (Figure 2.7b) or the identity (Figure 2.7c). On the other hand, since the activation function of the hidden units are required to be nonlinear, these can be only be chosen as the sigmoid or the hyperbolic tangent. The mathematical definition of these functions, and their corresponding derivatives are shown in Table 2.4.

Table 2.4: Mathematical definition and derivative of the activation functions

Activation function	**Mathematical definition**	**Derivative**
sigmoid	$f(\tau) = \dfrac{1}{1+e^{-\tau}}$	$f'(\tau) = \dfrac{e^{-\tau}}{(1+e^{-\tau})^2}$
hyperbolic tangent	$f(\tau) = \tanh(\tau)$	$f'(\tau) = \mathrm{sech}^2(\tau)$
identity	$f(\tau) = \tau$	$f'(\tau) = 1$

Given the input vector

$$\delta = \begin{bmatrix} \delta_1 & \delta_2 & \delta_3 & \cdots & \delta_I \end{bmatrix}^T \tag{2.27}$$

the transfer function of the feedforward neural network of Fig. 2.5 is given by the equations

$$\mathbf{h} = f_h(\mathbf{V}\delta') \tag{2.28a}$$

$$\zeta = f_\zeta(\mathbf{W}\mathbf{h}') \tag{2.28b}$$

where δ' is the modified input vector

$$\delta' = \begin{bmatrix} 1 & \delta_1 & \delta_2 & \cdots & \delta_I \end{bmatrix}^T \tag{2.29}$$

$\mathbf{h}$ is the vector of responses of the H units in the hidden layer

$$\mathbf{h} = \begin{bmatrix} h_1 & h_2 & h_3 & \cdots & h_J \end{bmatrix}^T \tag{2.30}$$

$\mathbf{h}'$ is the vector

$$\mathbf{h}' = \begin{bmatrix} 1 & h_1 & h_2 & \cdots & h_J \end{bmatrix}^T, \tag{2.31}$$

ζ is the output vector of the FFNN

$$\zeta = \begin{bmatrix} \zeta_1 & \zeta_2 & \zeta_3 & \cdots & \zeta_J \end{bmatrix}^T, \tag{2.32}$$

$\mathbf{V}$ and $\mathbf{W}$ are the matrices defined in (2.24), f_h is the activation function of the hidden units and f_ζ is the activation function of the output units.

c) Training algorithm

Different algorithms have been developed to train a FFNN. The backpropagation training algorithm was developed by Rummelhart *et al.* in 1986 [77], and boosted the research in feedforward neural networks. Given a set of S training vectors

$$\begin{array}{cccc} & \textbf{Input} & & \\ \delta_1^1 & \delta_2^1 & \cdots & \delta_I^1 \\ \delta_1^2 & \delta_2^2 & \cdots & \delta_I^2 \\ & \vdots & & \\ \delta_1^S & \delta_2^S & \cdots & \delta_I^S \end{array} \qquad \begin{array}{cccc} & \textbf{Output} & & \\ \zeta_1^1 & \zeta_2^1 & \cdots & \zeta_O^1 \\ \zeta_1^2 & \zeta_2^2 & \cdots & \zeta_O^2 \\ & \vdots & & \\ \zeta_1^S & \zeta_2^S & \cdots & \zeta_O^S \end{array}$$

for a FFNN $I \times J \times O$, the backpropagation algorithm uses the method of steepest descent to search for the weights that minimize a function of the network error. This gives rise to two versions of the algorithms, namely (i) off-line, in which the weights are updated after a complete pass through all the training vectors, i.e. the objective function to be minimized is given by

$$E = \frac{1}{2} \sum_{s=1}^{S} \sum_{o=1}^{O} \left(\zeta_o^s - \widehat{\zeta}_o^s \right)^2 \tag{2.33}$$

and (ii) on-line, in which the weights are updated after each training vector is presented, i.e. the aim is to minimize

$$E = \frac{1}{2} \sum_{o=1}^{O} \left(\zeta_o^s - \widehat{\zeta}_o^s \right)^2 \quad s = 1, \ldots, S. \tag{2.34}$$

In (2.33) and (2.34), superscript s identifies the training vector and subscript o refers to the output unit, ζ_o^s is the target value of the oth output in the sth training vector, and $\widehat{\zeta}_o^s$ is the network estimate for ζ_o^s. Note that any of the versions involves an iterative procedure to find successive approximations of the minimum, due to the significantly high dimensionality of the search space, given by $J \times (I+1) + O \times (J+1)$. The on-line version is described here, since the interest is to update the model after each measurement from the network becomes available.

According to the backpropagation algorithm, the weight w_{oj} is updated using the equation

$$w_{oj}(k+1) = w_{oj}(k) - \eta \frac{\partial E_s}{\partial w_{oj}} \quad o = 1, \ldots, O \quad j = 1, \ldots, J \tag{2.35}$$

where $w_{oj}(k)$ is the value of such weight in the kth training iteration, and η is a constant known as training rate. Now, from equations (2.28a) and (2.34), and applying the chain rule, it can be shown that $\frac{\partial E_s}{\partial w_{oj}}$ in (2.35) is given by

$$\frac{\partial E_s}{\partial w_{oj}} = -\left(\zeta_o^s - \widehat{\zeta}_o^s\right)\left[f_o'(\sigma)\mid_{\sigma=a_o}\right] h_j \tag{2.36}$$

where

$$a_o = \sum_{j=0}^{J} \zeta_o^s w_{oj} \tag{2.37}$$

and $f_o'(\sigma)$ is the derivative of the activation function of the output units (from Table 2.4).

On the other hand, the weight v_{ji} is updated by means of the equation

$$v_{ji}(k+1) = v_{ji}(k) - \eta \frac{\partial E_s}{\partial v_{ji}} \quad j = 1, \cdots, J \quad i = 1, \cdots, I \tag{2.38}$$

where $v_{ji}(k)$ is the value of the weight in the kth training iteration and η is the training rate. As before, from (2.28a) and (2.34), and applying the chain rule, it is obtained that $\dfrac{\partial E_s}{\partial v_{ji}}$ in (2.38) is given as

$$\frac{\partial E_s}{\partial v_{ji}} = -\sum_{o=1}^{O} \left(\zeta_o^s - \widehat{\zeta}_o^s\right) \left[f_o'(\sigma)\Big|_{\sigma=a_o} \right] w_{oj} \left[f_h'(\sigma)\Big|_{\sigma=b_j} \right] \delta_i^s \tag{2.39}$$

where

$$b_j = \sum_{i=0}^{I} \delta_i^s v_{ji} \tag{2.40}$$

and $f_h'(\sigma)$ is the derivative of the activation function of the hidden units (from Table 2.4).

As any gradient-based method, the backpropagation algorithm is susceptible to be trapped in local minima, thus it is suggested to carry out various trainings processes with different initial weights, and select the one that exhibits the best performance. As an alternative, different variants of the backpropagation algorithm [77] have been developed which use more efficient minimization approaches such as Newton method and conjugate gradient, among others. One of these variants is known as Levenberg-Marquardt, which has exhibited very good training performance.

As stated before, the training speed becomes a key issue when using a neural network to identify a time-varying model, as it is the case when describing traffic in wireless networks. In particular, the aforementioned algorithms may be rather slow because they may require several iterations before converging. Besides, their performance depends critically of some training parameters that should be specified by the user, and no clear rules exists for selecting appropriate values of these parameters for a particular application.

In order to overcome the training speed issue, in this work the neural networks are trained using the online sequential extreme learning algorithm (OS-ELM) [69], which is obtained as a recursive version of the extreme learning machine (ELM) [70]; the universal approximation capability of the OS-ELM is demonstrated in [78]. Extended least-squares ([68, 79]) (also called approximate maximum-likelihood method) is used to implement OS-ELM, instead of recursive least squares used in [69].

Consider again the above feedforward neural network with I input units, one layer of J hidden units, and O output units. For the OS-ELM algorithm, the activation function of the hidden and outputs units are the hyperbolic tangent and the identity functions, respectively. In other words, given the input vector $\delta = [\delta_1 \quad \delta_2 \quad \delta_3 \quad \cdots \quad \delta_I]^T$, the response of the jth hidden unit is calculated as

$$h_j = \rho\left(\mathbf{v}_j^T \delta'\right), \quad j = 1, \ldots, J \tag{2.41}$$

where $\rho(\cdot) = \tanh(\cdot)$ is the activation function for the hidden units and, as before, $\delta' = [1 \quad \delta_1 \quad \delta_2 \quad \cdots \quad \delta_I]^T$ and $\mathbf{v}_j = [v_{j0}, v_{j1}, \ldots, v_{jI}]^T$ is the weight vector for the jth hidden unit, where v_{j0} is the bias of such hidden unit, and v_{ji} is the weight connecting the ith input unit to the this hidden unit. Considering the identity activation function for the output units, the response of the oth output unit is given by

$$\widehat{\zeta}_o = \mathbf{h}'^T \mathbf{w}_o, \quad o = 1, \ldots, O \tag{2.42}$$

where $\mathbf{h}' = [1 \quad h_1 \quad h_2 \quad \cdots \quad h_J]^T$ and $\mathbf{w}_o = [w_{o0}, w_{o1}, \ldots, w_{oJ}]^T$ is the weight vector for the oth output unit, with w_{o0} the bias of such output unit, and w_{oj} the weight connecting the jth hidden unit to the this output unit.

After choosing the number J of hidden units, the OS-ELM algorithm involves two phases:

A) Initialization

An initial set of S_0 training vectors is used to implement the following initialization procedure:

(i) Randomly assign the weights between the input and hidden layers. Random assignment (with any distribution) of these weights guarantees that the approximation error can be made arbitrarily small ([68,69]).

(ii) Compute matrix

$$\mathbf{H}(0) = \begin{bmatrix} h_{11} & \cdots & h_{1J} \\ \vdots & \ddots & \vdots \\ h_{S_0 1} & \cdots & h_{S_0 J} \end{bmatrix} \tag{2.43}$$

of size $S_0 \times J$, containing the responses of the J hidden units for the training vectors in the initial set.

(iii) For each output unit, estimate the initial vector of weights connecting such unit and the hidden units as

$$\mathbf{w}_o(0) = \mathbf{P}(0)\mathbf{H}^T(0)\zeta_o(0), \tag{2.44}$$

where

$$\mathbf{P}(0) = \left[\mathbf{H}^T(0)\mathbf{H}(0)\right]^{-1} \tag{2.45}$$

and $\zeta_o(0)$ is the vector containing the target values for the oth output in the initial set.

(iv) Set $k = 1$.

B) Sequential learning

While there are more sets of data

(i) Pick the kth set with S_k training vectors.

(ii) Calculate the partial hidden layer output matrix $\mathbf{H}(k)$, of size $S_k \times J$, containing the responses of the J hidden units for the training vectors in kth set.

(iii) For each output unit, update the vector of weights connecting such unit and all hidden units using the equation

$$\mathbf{w}_o(k) = \mathbf{w}_o(k-1) + \frac{\mathbf{P}(k-1)\mathbf{H}^T(k)}{\alpha(k) + \mathbf{H}(k)\mathbf{P}(k-1)\mathbf{H}^T(k)} \left[\zeta_o(k) - \mathbf{H}(k)\mathbf{w}_o(k-1)\right] \tag{2.46}$$

where $\zeta_o(k)$ is the vector containing the target values for the oth output in the kth chunk,

$$\mathbf{P}(k) = \frac{\mathbf{P}(k-1)}{\alpha(k)} \left[\mathbf{I} + \frac{\mathbf{H}^T(k)\mathbf{H}(k)\mathbf{P}(k-1)}{\alpha(k) + \mathbf{H}(k)\mathbf{P}(k-1)\mathbf{H}^T(k)}\right] \tag{2.47}$$

and $\alpha(k)$ is a forgetting factor given by the difference equation

$$\alpha(k) = \alpha_o \alpha(k-1) + (1 - \alpha_o) \tag{2.48}$$

with initial condition $0.9 < \alpha(0) < 1$, and an updating factor $0 < \alpha_o < 1$. The use of a forgetting factor will give a better performance for time-varying systems. In addition, extended least squares is computationally more efficient since the update of $\mathbf{P}(k)$ requires no matrix inversion.

(iv) Set $k = k+1$.

Prior to the training, all entries of the input and target vectors in the training set are linearly normalized to span the interval $[-1, 1]$, using the formula

$$\theta = -1 + 2\frac{\Theta - \Theta_{\text{min}}}{\Theta_{\text{max}} - \Theta_{\text{min}}} \tag{2.49}$$

where Θ and θ represent the original and normalized values, respectively, while Θ_{min} and Θ_{max} are the corresponding lower and upper bounds, respectively. Using this normalization procedure, the neural network showed good learning performance with the random weights adjusted using a Gaussian distribution with zero mean and standard deviation one.

From (2.49), the normalization function

$$\theta = \text{nor}(\Theta) = \frac{2}{\Theta_{\text{max}} - \Theta_{\text{min}}}\Theta - \frac{\Theta_{\text{max}} + \Theta_{\text{min}}}{\Theta_{\text{max}} - \Theta_{\text{min}}} \tag{2.50}$$

and the denormalization function

$$\Theta = \text{dnor}(\theta) = \frac{\Theta_{\text{max}} - \Theta_{\text{min}}}{2}\theta + \frac{\Theta_{\text{max}} + \Theta_{\text{min}}}{\Theta_{\text{max}} - \Theta_{\text{min}}} \tag{2.51}$$

can be defined [68]. Then, the denormalized (original) value of the oth output of the neural network is given by

$$\widehat{\mathbf{Z}}_o = \text{dnor}(\widehat{\zeta}_o) = \text{dnor}(\mathbf{h}'^T \mathbf{w}_o, \quad o = 1, \ldots, O. \tag{2.52}$$

Then, in the above definitions of δ and δ', δ_i $(i = 1, \ldots, I)$ can be equivalently expressed as

$$\delta_i = \text{nor}(\Delta_i), \quad i = 1, \ldots, I \tag{2.53}$$

where Δ_i is the original value of the ith input.

In summary, since the the neural network model is used to estimate original values of the outputs from original values of the inputs, the nonlinear input-output relationship implemented by the neural network model can be described by

$$\widehat{\mathbf{Z}}_o = \text{dnor}(\mathbf{h}^T \mathbf{w}_o), \quad o = 1, \ldots, O \tag{2.54a}$$

where $\mathbf{w}_o = [w_{o0}, w_{o1}, \ldots, w_{oJ}]^T$ is the weight vector for the oth output unit, with w_{o0} the bias and w_{oj} the weight connecting the jth hidden unit to this output unit, and $\mathbf{h}' = [1, h_1, \ldots, h_J]^T$ with

$$h_j = \rho(\mathbf{v}_j^T \delta'), \quad j = 1, \ldots, J \tag{2.54b}$$

where $\rho(\cdot) = \tanh(\cdot)$ is the activation function for the hidden units, $\mathbf{v}_j = [v_{j0}, v_{j1}, \ldots, v_{jI}]^T$

is the weight vector for the jth hidden unit, where v_{j0} is the bias and v_{ji} is the weight connecting the ith input unit to this hidden unit, and $\delta' = [1, \mathrm{nor}(\Delta_1), \ldots, \mathrm{nor}(\Delta_I)]^T$.

2.4.2.2.3 Optimal linearization of a FFNN-based model

The neural network nonlinear model defined by (2.54) can be written in a simplified form as

$$\widehat{\mathbf{Z}}_o = f_o(\mathbf{\Delta}), \quad o = 1, \ldots, O \tag{2.55}$$

where $\mathbf{\Delta} = [\Delta_1, \ldots, \Delta_I]^T$, and $f_o : \mathbb{R}^I \rightarrow \mathbb{R}$ is the nonlinear function that calculates the oth output. Given a particular input point $\mathbf{\Delta} = \overline{\mathbf{\Delta}}$, it is desired to find a linear model of the form

$$\widehat{\mathbf{Z}}_o = \mathbf{G}^T \mathbf{\Delta}, \quad o = 1, \ldots, O \tag{2.56}$$

with $\mathbf{G} = [\gamma_1, \ldots, \gamma_I]^T$, which is locally equivalent to (2.55)) around $\overline{\mathbf{\Delta}}$. Taylor's linearization approach, which has been the most commonly used local linearization technique, is not applicable in the vicinity of any input point of the system. Even if such input point is an equilibrium point, this technique will yield an affine rather than a linear model if such equilibrium is not the origin [68]. In order to overcome this drawback, the optimal linearization approach proposed by Teixeira and Żak [80] is used for local linearization of the neural network model. This approach is now described [68].

Equation (2.55) can be approximated around $\mathbf{\Delta} = \overline{\mathbf{\Delta}}$ by a Taylor series expansion, i.e.

$$\widehat{\mathbf{Z}}_o = f_o(\mathbf{\Delta}) \approx f_o(\overline{\mathbf{\Delta}}) + \nabla_{\Delta^T} f_o(\overline{\mathbf{\Delta}})[\mathbf{\Delta} - \overline{\mathbf{\Delta}}] = f_o(\overline{\mathbf{\Delta}}) + \nabla_{\Delta^T} f_o(\overline{\mathbf{\Delta}}) \cdot \mathbf{\Delta} - \nabla_{\Delta^T} f_o(\overline{\mathbf{\Delta}}) \cdot \overline{\mathbf{\Delta}} \quad (2.57)$$

where $\nabla_{\Delta^T} f_o(\overline{\mathbf{\Delta}})$ is the Jacobian of $f_o(\mathbf{\Delta})$ evaluated at $\mathbf{\Delta} = \overline{\mathbf{\Delta}}$. Note that, for the approximation to be linear

$$f_o(\overline{\mathbf{\Delta}}) - \nabla_{\Delta^T} f_o(\overline{\mathbf{\Delta}}) \cdot \overline{\mathbf{\Delta}} = 0 \quad (2.58)$$

for which the linear model is given by

$$\widehat{\mathbf{Z}}_o = \mathbf{G}^T \mathbf{\Delta} \quad (2.59)$$

with $\mathbf{G}^T = \nabla_{\Delta^T} f_o(\overline{\mathbf{\Delta}})$. Now, (2.58) holds only if $\overline{\mathbf{\Delta}}$ is the origin and if that origin is an equilibrium point. Therefore, even though Taylor linearization is not applicable in the vicinity of any operation point of the system because in general (2.58) does not hold, it is the linearization method most commonly used with the linear model calculated according to (2.59), thus resulting in a degradation of the quality of this linear approximation.

Then, according to the optimal linearization approach, given a particular operating point $\mathbf{\Delta} = \overline{\mathbf{\Delta}}$ it is desired to find a linear model of the form

$$\widehat{\mathbf{Z}}_o = \mathbf{G}^T \mathbf{\Delta}, \quad o = 1, \cdots, O \quad (2.60)$$

such that, at the operating point

$$f_o(\overline{\mathbf{\Delta}}) = \mathbf{G}^T\overline{\mathbf{\Delta}} \tag{2.61}$$

and in the vicinity of the operating point

$$f_o(\mathbf{\Delta}) \approx \mathbf{G}^T\mathbf{\Delta} \tag{2.62}$$

according to the least squares criterion, where $\mathbf{\Delta}$ represents a point close to $\overline{\mathbf{\Delta}}$.

As stated before, the Taylor series approximation of $f_o(\mathbf{\Delta})$ around $\overline{\mathbf{\Delta}}$ is given by

$$f_o(\mathbf{\Delta}) \approx f_o(\overline{\mathbf{\Delta}}) + \nabla_{\Delta^T} f_o(\overline{\mathbf{\Delta}})[\mathbf{\Delta} - \overline{\mathbf{\Delta}}] \tag{2.63}$$

where $\nabla_{\Delta^T} f_o(\overline{\mathbf{\Delta}})$ is the Jacobian of $f_o(\mathbf{\Delta})$ evaluated at $\mathbf{\Delta} = \overline{\mathbf{\Delta}}$. Considering (2.61) and (2.62), equation (2.63) can be rewritten as

$$\mathbf{G}^T\mathbf{\Delta} \approx \mathbf{G}^T\overline{\mathbf{\Delta}} + \nabla_{\Delta^T} f_o(\overline{\mathbf{\Delta}})[\mathbf{\Delta} - \overline{\mathbf{\Delta}}] \tag{2.64}$$

which is equivalent to

$$\mathbf{G}^T[\mathbf{\Delta} - \overline{\mathbf{\Delta}}] \approx \nabla_{\Delta^T} f_o(\overline{\mathbf{\Delta}})[\mathbf{\Delta} - \overline{\mathbf{\Delta}}]. \tag{2.65}$$

In order for (2.65) to be satisfied, it must hold that

$$\mathbf{G}^T \approx \nabla_{\Delta^T} f_o(\overline{\boldsymbol{\Delta}}) \tag{2.66}$$

which gives rise to the objective function

$$\Gamma = \frac{1}{2}\left\|\mathbf{G}^T - \nabla_{\Delta^T} f_o(\overline{\boldsymbol{\Delta}})\right\|_2^2. \tag{2.67}$$

The optimal linear model can be determined minimizing (2.67), subject to the constraint (2.61).

According to the method of Lagrange multipliers, the augmented function to be minimized is given by

$$\Gamma_a = \frac{1}{2}\left\|\mathbf{G}^T - \nabla_{\Delta^T} f_o(\overline{\boldsymbol{\Delta}})\right\|_2^2 + \lambda\,[f_o(\overline{\boldsymbol{\Delta}}) - \mathbf{G}^T\overline{\boldsymbol{\Delta}}] \tag{2.68}$$

where λ is a scalar Lagrange multiplier that can be determined by setting the gradients of Γ_a with respect to $\mathbf{G}^T$ and λ equal to zero, and solving for $\mathbf{G}^T$ to obtain

$$\mathbf{G}^T = \nabla_{\Delta^T} f_o(\overline{\boldsymbol{\Delta}}) + \frac{f_o(\overline{\boldsymbol{\Delta}}) - \nabla_{\Delta^T} f_o(\overline{\boldsymbol{\Delta}})\overline{\boldsymbol{\Delta}}}{\overline{\boldsymbol{\Delta}}^T\overline{\boldsymbol{\Delta}}}\overline{\boldsymbol{\Delta}}^T \tag{2.69}$$

where

$$\nabla_{\Delta^T} f_o(\overline{\boldsymbol{\Delta}}) = \left[\frac{\partial f_o}{\partial \Delta_1}(\overline{\boldsymbol{\Delta}}), \cdots, \frac{\partial f_o}{\partial \Delta_I}(\overline{\boldsymbol{\Delta}})\right] \tag{2.70}$$

where (according to (2.54) and (2.57))

$$\frac{\partial f_o}{\partial \Delta_i} = \frac{d}{d\theta}\mathrm{dnor}_o(\theta) \times \sum_{j=1}^{J} \left\{ w_{oj} \cdot \left[\frac{d}{d\beta}\rho(\beta) \right] \bigg|_{\beta=\overline{\beta}} \cdot v_{ji} \cdot \frac{d}{d\Theta}\mathrm{nor}_i(\Theta) \right\} \tag{2.71}$$

where

$$\overline{\beta} = \sum_{i=0}^{I} v_{ji} \cdot \mathrm{nor}_i(\overline{\Delta}_i), \tag{2.72}$$

and, from (2.50) and (2.51)

$$\frac{d}{d\Theta}\mathrm{nor}_i(\Theta) = \frac{2}{\Theta_{\max_i} - \Theta_{\min_i}} \tag{2.73}$$

and

$$\frac{d}{d\theta}\mathrm{dnor}_o(\theta) = \frac{\Theta_{\max_o} - \Theta_{\min_o}}{2}. \tag{2.74}$$

In (2.73)) and (2.74), $\Theta_{\max_i}$ and $\Theta_{\min_i}$ represent the upper and lower bounds, respectively, for the ith input, and $\Theta_{\max_o}$ and $\Theta_{\min_o}$ represent the upper and lower bounds, respectively, for the oth output. The optimal linear model will have exactly the same dynamics of the original nonlinear model at the input point, and minimum modeling error in the neighborhood of such point [68].

Note that if $\overline{\boldsymbol{\Delta}}^T\overline{\boldsymbol{\Delta}} = 0$, equation (2.69) becomes

$$\mathbf{G}^T = \nabla_{\Delta^T} f_o(\overline{\boldsymbol{\Delta}}) \tag{2.75}$$

i.e., the optimal linearization approach reduces to Taylor's linearization.

2.4.2.2.4 Nonlinear modeling approaches based on feedforward neural networks and linear Kalman innovation models

The linear Kalman innovation model described previously may not exhibit a good modeling performance when describing nonlinear phenomena in wireless communication networks. Nonlinear modeling approaches become an alternative to overcome this limitation, especially those based on neural networks, since they have universal approximation capabilities [67].

2.4.2.2.4.1 NN-ARX

Aljerme and Liu [51] presented a discrete-time time-varying neural-network based autoregressive with exogenous inputs (NN-ARX) model of the traffic in mobile telephony networks, which is identified and continuously updated.

In the case of a NN-ARX model of a process, a neural network is trained to estimate the output $y(k+1)$ at time instant $k+1$, as a nonlinear function of p previous values of the output and input, i.e.

$$\hat{y}(k+1) = F^{(k)}\Big[y(k),\cdots,y(k-p+1),u(k),\cdots,u(k-p+1)\Big] \tag{2.76}$$

where $y(k)$ and $u(k)$ are, respectively, the output and input at time instant k, F denotes the nonlinear function implemented by the neural network (equation (2.54)), and the superscript (k) indicates that the NN-ARX model has been updated using the data vectors up to the kth sample. Note that the neural network corresponding to the NN-ARX model (2.76) has one output and $2p$ inputs.

As shown later in the Results section, the NN-ARX model (2.76) exhibited good performance in describing traffic in a mobile telephony network. However, as stated before, it should be remarked that this model cannot be transformed into a Kalman model, which limits its application in control tasks.

2.4.2.2.4.2 NN-ARMAX

As an alternative, it can be used a discrete-time time-varying neural network-based autoregressive moving average with exogenous inputs (NN-ARMAX) model, which may be converted into a locally equivalent linear time-varying Kalman innovation model, after optimally linearizing the NN-ARMAX at the current operating point.

A NN-ARMAX model of a process is built to estimate $y(k+1)$ as a nonlinear function of p previous values of the output, input and innovation error, i.e.

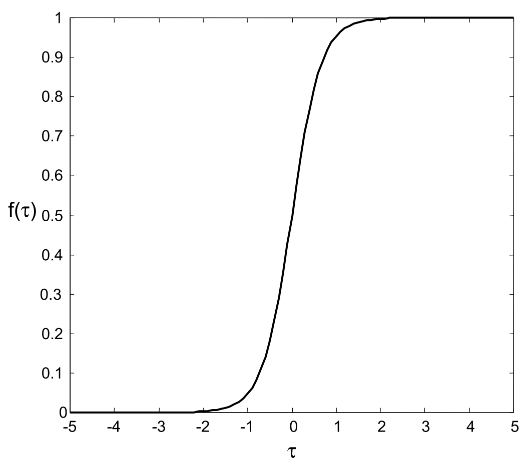
$$\hat{y}(k+1) = F^{(k)}\Big[y(k),\cdots,y(k-p+1),u(k),\cdots,u(k-p+1)$$
$$e(k),\cdots,e(k-p+1)\Big] \tag{2.77}$$

where $e(k)$ is the innovation error at time instant k. Compared to an NN-ARX model, it can be seen that the NN-ARMAX model has p previous values of the innovation error as additional inputs, thus resulting in a total of $3p$ inputs. These additional inputs enable constructing a linear Kalman innovation model applying the following procedure for each training data vector [68]: (i) update the NN-ARMAX model by executing a new iteration of the OS-ELM algorithm, (ii) carry out the optimal linearization at the data point to obtain a locally equivalent linear ARMAX model and (iii) construct the linear time-varying Kalman innovation model, from the linear ARMAX model obtained in the previous step.

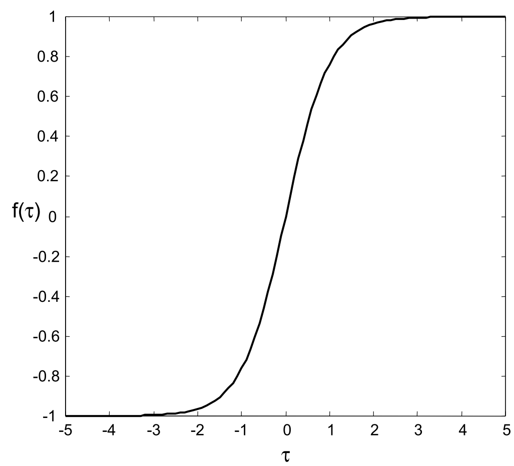
Step (i) involves executing the OS-ELM algorithm for each data vector that becomes available, while step (iii) entails the same process explained above, through which the linear Kalman innovation model (2.23) is constructed from the linear ARMAX model (2.18).

Now, according to step (ii), the linear ARMAX model of the form given by (2.18) is obtained by optimal linearization of the NN-ARMAX model (2.77) at the current input point [68].

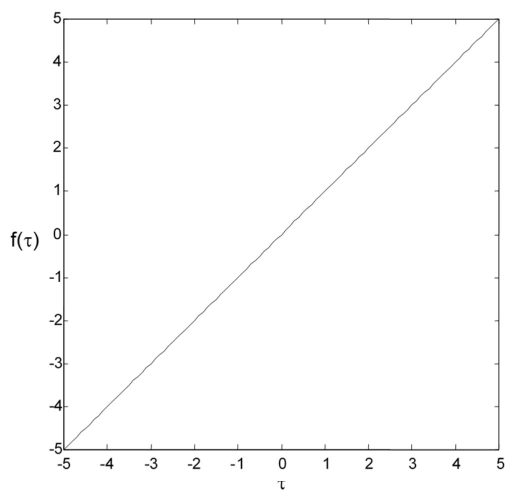
Now, it is important to remark that any neural network-based model, including NN-ARX and NN-ARMAX, are black-box model, i.e. their parameters do not have physical meaning in terms of the process they are describing, which in this work is the dynamic of wireless networks. Nevertheless, this does not limit their application in optimization and control approaches, in which it is mainly required to describe the input-output relationship.



(a)



(b)



(c)

Figure 2.7: Common activation functions: (a) sigmoid, (b) hyperbolic tangent and (c) identity.

Chapter 3: METHODOLOGY

3.1 First case study: linear Kalman innovation modeling in indoor multipath wireless channels

This case study evaluates the performance of linear Kalman innovation models in observable block companion form [68], for describing the dynamics of measured Received Signal Strength Indicator (RSSI) in indoor multipath wireless channels [48].

As stated before, multipath occurs when the transmitted signal arrives at the receiver through multiple propagation paths, i.e., the direct and/or reflection paths. The multipath phenomenon appears in wireless environments with complex geometry, in which the wave reflects on large enough uniform surfaces, and the reflected ray is coherent and produces inter-symbol interference (ISI) with the ray in the line of sight. Each path may have a different phase, attenuation, delay and Doppler frequency [48]. The multipath channels may be of two types: (i) Rayleigh, which considers no line of sight between transmitter and receiver nodes, i.e. only echoes of the original signal are received, and (ii) Ricean, in which the signal comprises the ray in the line of sight plus delayed and attenuated versions of it [48].

Specifically, this case study involves the identification of a model that describes the Received Signal Strength Indicator (RSSI, output) as a function of the speed of the link (LinkSpeed, input) [48]. The RSSI is related to the power of the signal and enables to distinguish one channel with multipath from another.

The measuring process was conducted by the application G-NetWiFi Pro, in a channel with multipath constituted by an indoor environment, specifically a single-story housing, with obstacles sufficiently uniform and of size larger than the wave length. The transmitter node (Alice) is a WiFi router with protocol 5.8 GHz 802.11/g, while the receiver node (Bob) corresponds to a smartphone being used to download and watch a movie (heavy file) in real time, and simultaneously run G-NetWiFi Pro [48].

Figure 3.1 illustrates the floor plan of the housing, in which the WiFi router is at a fixed location, while the smartphone is moved around and placed at different positions, guaranteeing that there are many obstacles between Alice and Bob and that there is no line of sight between them. Therefore, the following conditions are met during the measurement process: (i) the channel is of Rayleigh type, and (ii) the RSSI, given in dBm, and the LinkSpeed, in Megabits per second (Mbps), are measured at the Bob end, with a sampling time of 1 sec; this sampling time is appropriate according to the variability of the signal. Figures 3.2a and 3.2b plot the 1561 measurements of RSSI and LinkSpeed, respectively [48].

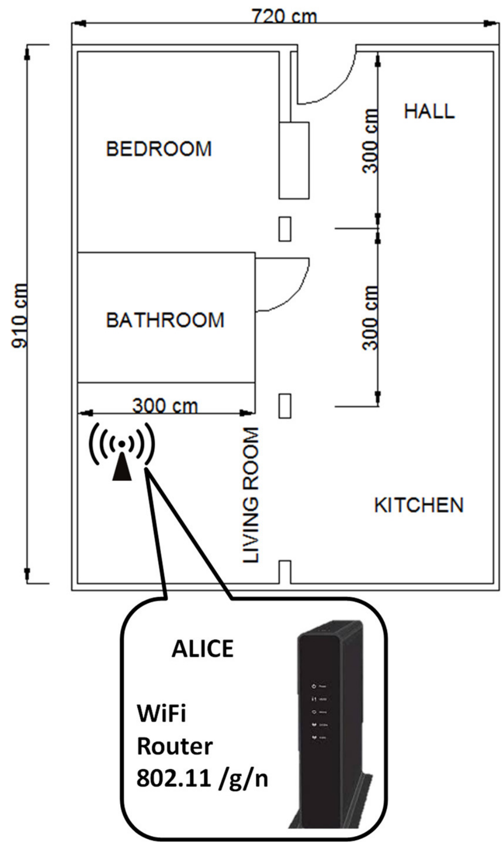


Figure 3.1: Floor plan of the housing and location of the WiFi router during the measurement procedure [48].

Considering RSSI as the output and LinkSpeed as the input, the procedure described in chapter II is used to construct a linear Kalman Innovation model of the form (2.23), varying the number of previous values of the output, input and innovation error.

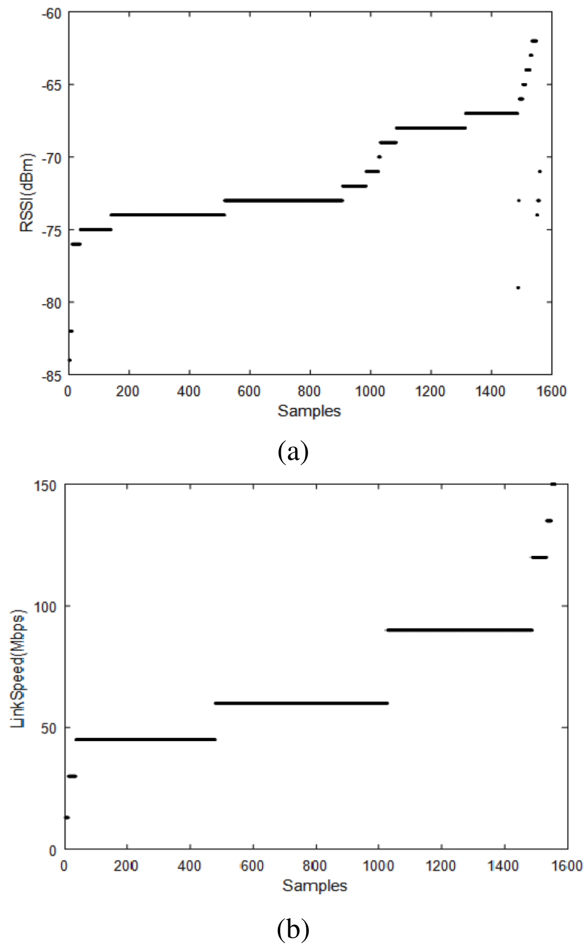


Figure 3.2: Plot of measured values of (a) RSSI, (b) LinkSpeed [48].

3.2 Second case study: secret key generation in the physical layer in multipath wireless channels

A positive capacity for key generation can be achieved if the correlation between the measurements of the legitimate users (Alice and Bob) is higher than the correlation between measurements of a legitimate user and an eavesdropper (Eve). This guarantees that Eve will not be able to get the secret key, unless Eve has the same location as Alice or Bob [4].

As it was described before, the algorithm for generating secret keys in the physical layer involves six steps [60]: (i) Initialization or beacon exchange, (ii) estimation of the common source of randomness, (iii) quantization, (iv) encoding, (v) information reconciliation and (vi) privacy amplification.

Framed in step 2, this case study proposes the use of linear time-varying Kalman innovation models, identified from real RSSI measurements, for secret key generation in multipath wireless channels. The multipath introduces randomness because the trajectories of the reflected rays depend on the geometry of the channel between Alice and Bob. As a result, the channel between any pair of communicating users is unique, and this feature is exploited for secret key generation. Specifically, after the identification has converged, the residuals will be highly random and can be utilized to generate secret keys, as will be shown later. As stated before, RSSI is used because CSI has a significantly greater computational cost, which hampers its possibility of fulfilling the requirement that each key must be generated quickly and renewed frequently [47].

Successive blocks of RSSI measurements are generated during the conversation of Alice and Bob, and the key should be renewed at the end of each block to limit the amount of unsecure information in case the key is correctly guessed by an eavesdropper. These RSSI measurements are used by Alice and Bob to identify separate linear time-varying Kalman innovation models in observable block companion form of the multipath channel between them; the order P of these models is given by [5]

$$P = \lceil 2Mf_dT \rceil + 5 \tag{3.1}$$

where $\lceil 2Mf_dT \rceil$ denotes the smallest integer greater than $2Mf_dT$.

Now, the linear Kalman innovation models used here are a variant of the ones presented in section II, as explained in the following [47]. After the true RSSI measurement $g^X(k+1)$ at time instant $k+1$ is obtained, a linear autoregressive moving average (ARMA) model of the form

$$\begin{aligned} \hat{g}^X(k+1) &= -a^X_{1,k}g^X(k) - \cdots - a^X_{P,k}g^X(k-P+1) \\ &+ d^X_{1,k}e^X(k) + \cdots + d^X_{P,k}e^X(k-P+1) \end{aligned} \tag{3.2}$$

where X is A for Alice and B for Bob, is recursively updated according to the extended least squares algorithm with

$$\theta^X_k = \left[\begin{array}{ccccc} a^X_{1,k} & \cdots & a^X_{P,k} d^X_{1,k} & \cdots & d^X_{P,k} \end{array} \right]$$

the vector of model parameters, and

$$\varphi^X_{k+1} = \left[\begin{array}{ccccc} g^X(k) & \cdots & g^X(k-P+1) e^X(k) & \cdots & e^X(k-P+1) \end{array} \right]$$

the data vector. Then, the linear time-varying Kalman innovation model of the channel is given by [47]

$$\mathbf{x}_o^X(k+1) = \mathbf{A}_k^X \mathbf{x}_o^X(k) + \mathbf{K}_k^X \mathbf{e}^X(k) \tag{3.3a}$$

$$\hat{g}^X(k) = \mathbf{C}\mathbf{x}_o^X(k) \tag{3.3b}$$

where $\mathbf{x}_o^X(k) \in \mathbb{R}^{P\times 1}$, $\mathbf{e}^X(k) \in \mathbb{R}$ and $\hat{g}^X(k) \in \mathbb{R}$ are the observed state vector, the innovation error vector and the estimated output, respectively, at time instant k,

$$\mathbf{A}_k^X = \begin{bmatrix} -a_{1,k}^X & 1 & 0 & \cdots & 0 \\ -a_{2,k}^X & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{P,k}^X & 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{P\times P},$$

$$\mathbf{K}_k^X = \begin{bmatrix} d_{1,k}^X - a_{1,k}^X \\ d_{2,k}^X - a_{2,k}^X \\ \vdots \\ d_{P,k}^X - a_{P,k}^X \end{bmatrix} \in \mathbb{R}^{P\times 1}$$

and $\mathbf{C}$ is fixed given by

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \mathbb{R}^{1\times P}.$$

When calculating $\mathbf{x}_o^X(k+1)$, $\mathbf{e}^X(k)$ is the innovation error. This procedure is repeated for every measurement.

The measurement process was conducted on the single-story housing of Figure 3.3.

However, in this experiment the Alice node is a smart phone used as a WiFi router with protocol 2.4 GHz 802.11/g, and the Bob node is a Gw Instek 9 Khz$\sim$ 3 GHz spectrum analyzer model GSP-830 with preamplifier, and a TP link omnidirectional antenna with a gain of 2.15 dB; in addition there is a laptop with appropriate software and a USB interconnection cable [47].

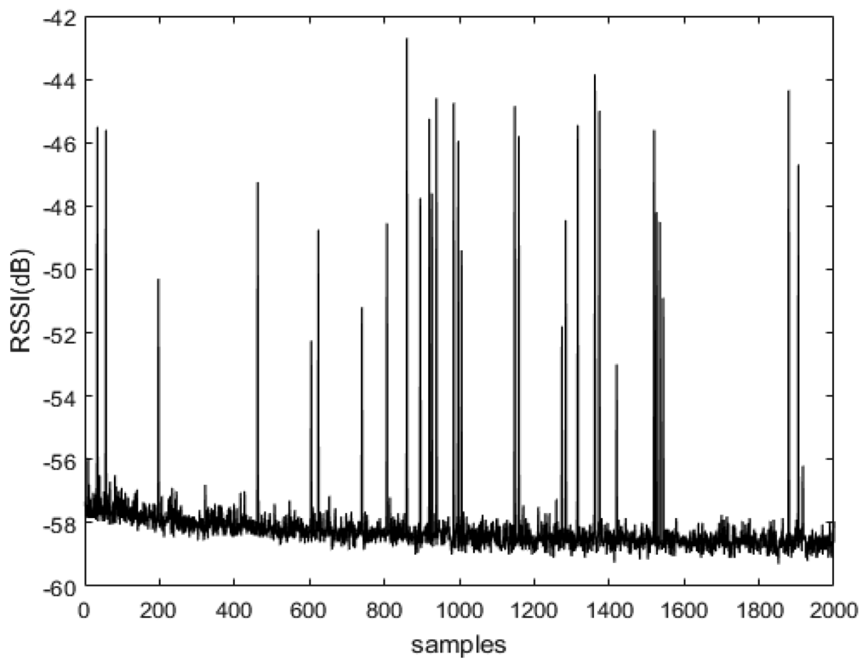


Figure 3.3: Measurements of RSSI in the Rayleigh Channel [47].

The following assumptions were made during the measurement process [47]: (i) channel is Rayleigh and geometrically complex, (ii) RSSI is measured only in one end, and alternately assigned to Alice and Bob (since the measurements taken from any end should be the same), and (iii) measurements are taken in discrete-time (through impulses). It was taken one block of $M = 2000$ measurements, which are plotted in Figure 3.3 [47].

Then, two Kalman models of the unique channel between Alice and Bob are simultaneously updated. Once the Kalman models have captured the dynamics of the multipath channel, the innovation errors are suitable for key generation. The process is reinitialized at the beginning of each block of measurements, and the innovation errors of the subsequently identified Kalman models are used for key renewal purposes [47]. The applicability of the innovation error for secret key generation is analyzed by means of tests such as randomness and correlation between the errors of Alice and Bob for the same block. In addition, the maximum capacity of the channel for key generation is determined as a function of SNR [47].

3.3 Third case study: time-varying neural network-based modeling schemes for describing traffic in mobile telephony networks

In this application, NN-ARX [51] and NN-ARMAX models are identified for describing traffic in mobile telephony networks. The available data corresponds to real traffic data taken from an LTE (4G) network in a populated city, in particular data from two stations (one cell per station) identified as A and B.

The data were collected with a sampling time of 1 hour, thus there are 24 samples in one day, and the total data span approximately 16 weeks. The output variable was the payload (given in Mbps), while the input variable was the number of users. The total number of data for station A was 2873 samples, with 2155 being used for training and 718 for validation. Regarding station B, a total of 2841 samples were available, with 2143 and 698 used for training and validation, respectively [51]. The training set is utilized to identify and recursively update the models, and the validation set is employed to test the performance.

3.4 Performance criteria

The criterion utilized to test the modeling performance of the identified models (linear Kalman innovation, NN-ARX and NN-ARMAX), is the percentage of data samples with relative errors smaller than a certain bound bnd. This percentage will be denoted as $\%bndT$ or $\%bndV$ for the training and validation samples, respectively, and will be calculated and plotted as a function of the number p of previous values. For the k-th data point, the relative error is calculated as

$$e_{rel}(k) = \frac{g(k) - \hat{g}(k)}{g(k)} \times 100 \tag{3.4}$$

Then, for the value of p giving the best performance on the validation set, a plot of the relative error for the data samples in such set is included.

On the other hand, regarding the application of secret key generation, RunsTest [81] is used to test the randomness of the residuals, and the correlation between the residuals of Alice and Bob for the same block is determined through the correlation coefficient. In addition, the maximum channel capacity for secret key generation, which is a measure of the maximum velocity at which the channel can generate secret keys, is calculated as [5]

$$B = \frac{1}{2}\log_2\left(\frac{|C_{AA}|\,|C_{BB}|}{\det(C)}\right) \tag{3.5}$$

given in bits/seconds/Hz, where

$$C = \mathrm{cov}(\theta_k^A, \theta_k^B) = \begin{bmatrix} C_{AA} & C_{AB} \\ C_{BA} & C_{BB} \end{bmatrix} \tag{3.6}$$

is the covariance matrix between θ_k^A and θ_k^B.

Chapter 4: RESULTS

4.1 First case study [48]

Using the 1561 available samples of RSSI and LinkSpeed, linear ARMAX models of the form (1) were identified and recursively updated, with p ranging from 1 to 20. Then, for each p, all samples of RSSI in the data were estimated using the corresponding linear Kalman innovation model at the final data sample. Subsequently, the percentage of data samples with a relative error smaller than 15% as a function of p was calculated and plotted (Figure 4.1). The best performance was obtained for $p = 1$, for which 99.87% of the samples fulfilled the criterion.

Figure 4.2 plots the relative error for all samples of RSSI in the data for $p = 1$. The corresponding Kalman model obtained at the last sample is [48]

$$\mathbf{x}_o(k+1) = 0.86\mathbf{x}_o(k) - 0.07u(k) + 0.83e(k) \tag{4.1a}$$

$$\hat{g}(k) = \mathbf{x}_o(k) \tag{4.1b}$$

Note that the relative error decreases as the sample number increases. This occurs because the Kalman model corresponding to the last sample was used to generate such

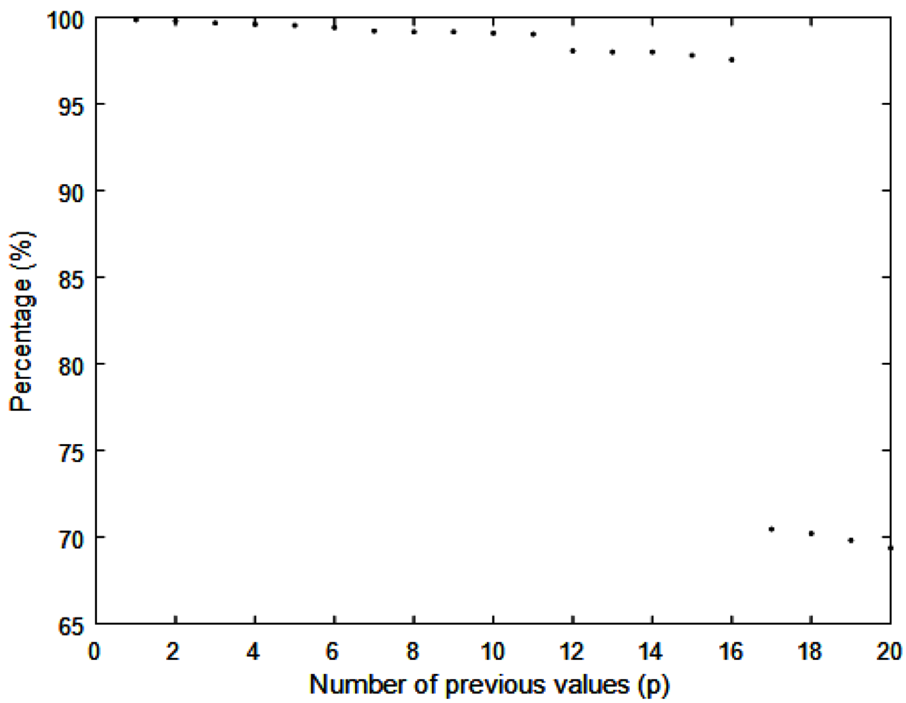


Figure 4.1: Plot of %15 vs. p [48].

curve, and due to the forgetting factor it is expected that the model behaves better for more recent data. This is completely desirable in a practical application, because more recent data is a more reliable representation of the current dynamics of the system [48].

Also note that the model exhibits robustness, since the percentage of samples of RSSI with errors below 15% is above 95% for $p \leq 16$.

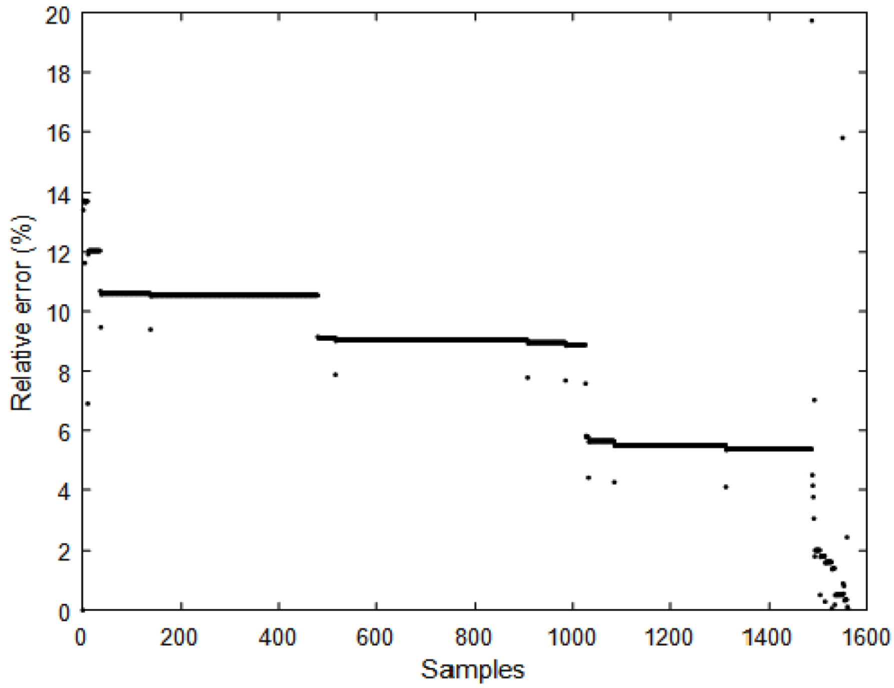
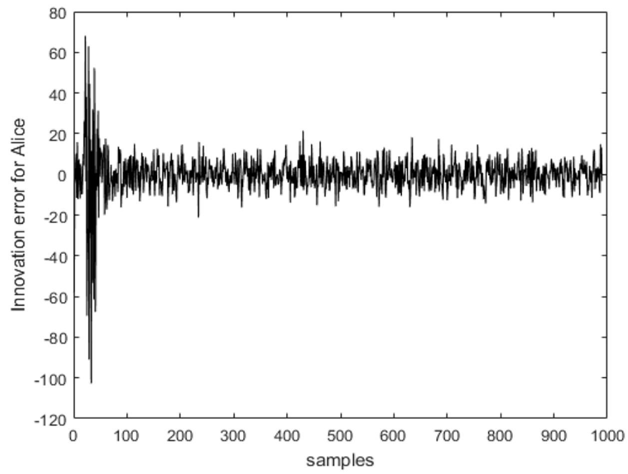


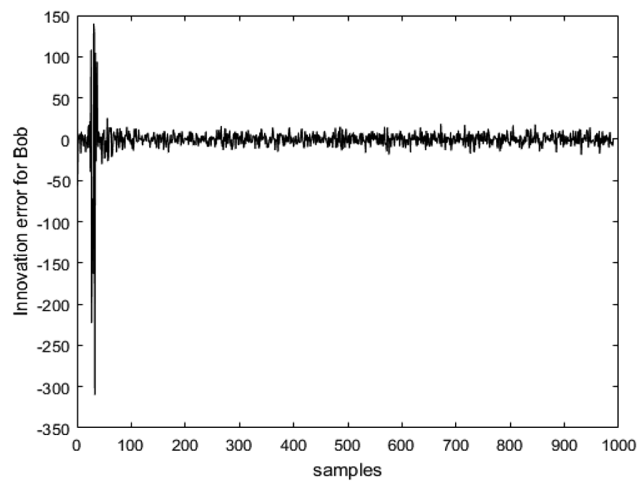
Figure 4.2: Relative error for the 1561 samples of RSSI for $p = 1$ [48].

4.2 Second case study [47]

Kalman models of order $P = 45$ were identified for Alice and Bob. Figures 4.3a and 4.3b show plots of the innovations sequences for Alice and Bob, respectively. The Kalman models converge approximately at sample 50 for both Alice and Bob, time after which the secret keys can be generated. Performance criteria and channel capacity were evaluated after convergence.



(a)



(b)

Figure 4.3: Innovation sequence for (a) Alice and (b) Bob, for $M = 2000$ [47].

First, RunstTest of the NIST standard indicated that both set of residuals are random with a significance level of 95%. Therefore, the randomness requirement is fulfilled.

On the other hand, the correlation coefficient between the residuals of Alice and Bob for the same block is 0.98, thus verifying that the high correlation requirement is also satisfied.

At last, the maximum channel capacity B for secret key generation is calculated as a function of signal-to-noise (SNR) ratio. Figure 4.4 plots discrete values of B vs. SNR, for SNR 0 dB, 5 dB, 10 dB, 15 dB and 20 dB, and also plots a straight line fitted using least-squares. The minimum and maximum values were 0.87 bps/Hz for 0 dB, and 2.82 bps/Hz for 15 dB, respectively. On the other hand, the line has a positive slope, with approximate extreme values of 0.8 and 2.87 for 0 dB and 20 dB, respectively. As a comparison, under similar conditions McGuire [5] obtained values of B between 0.07 for 0 dB and 0.18 for 20 dB. Therefore, the maximum channel capacity for the Kalman innovation model was approximately sixteen times greater [47].

4.3 Third case study

The procedure described in chapter II was used to identify NN-ARX models of the form given by (2.76) [51], and NN-ARMAX models of the form (2.77), using the data available for stations A and B. For both cases, the number p of previous values ranged from 1 to 20.

Table 4.1: Performance of the NN-ARX and NN-ARMAX models on the training data sets, for stations A and B

	NN-ARX		NN-ARMAX	
Station	%10 T	P	%10 T	P
A	68.9	2	65.3	1
B	65.0	2	66.2	1

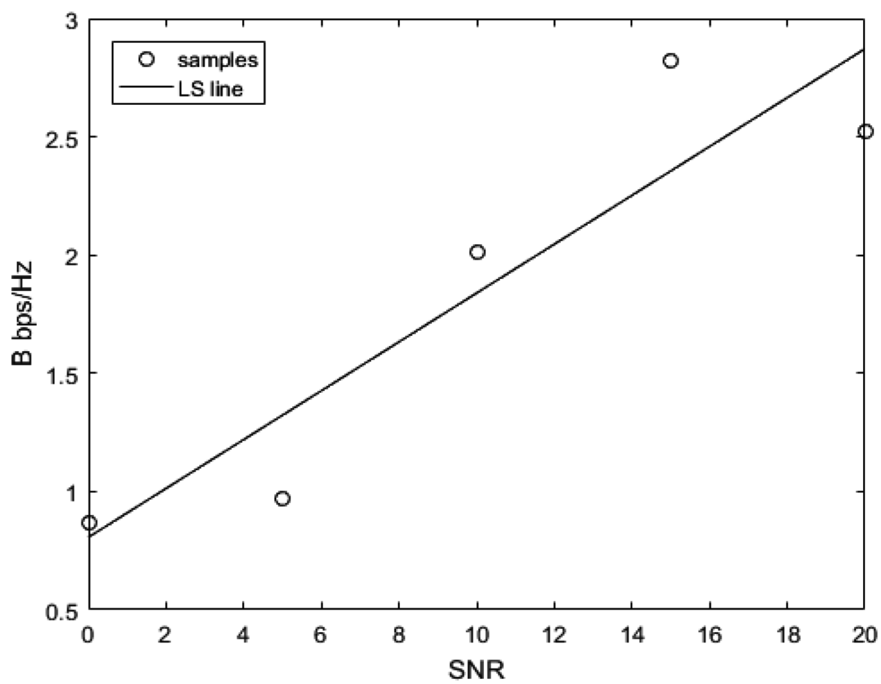


Figure 4.4: Plot of B vs. SNR [47].

The values of $\%10T$ are included in Table 4.1. Note that the performance of the two models is very similar for both stations, with the NN-ARX performing slightly better for station A and the NN-ARMAX for station B. For the two stations, the NN-ARX model exhibits the best performance for $p = 2$, while the NN-ARMAX model yields the best behavior for $p = 1$.

On the other hand, Table 4.2 shows the values of $\%10V$. As before, the performance of the two models is very similar, but the NN-ARMAX performs slightly better for both stations. Again, the NN-ARX and NN-ARMAX models exhibit the best performance for

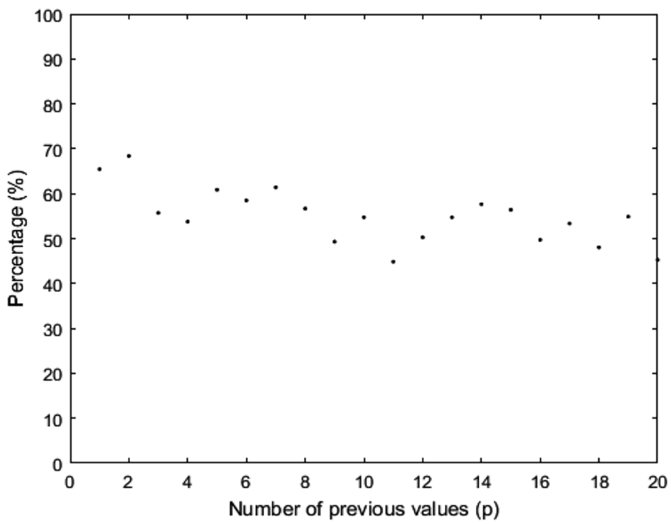
$p = 2$ and for $p = 1$, respectively, for the two stations.

Table 4.2: Performance of the NN-ARX and NN-ARMAX models on the validation data sets, for stations A and B

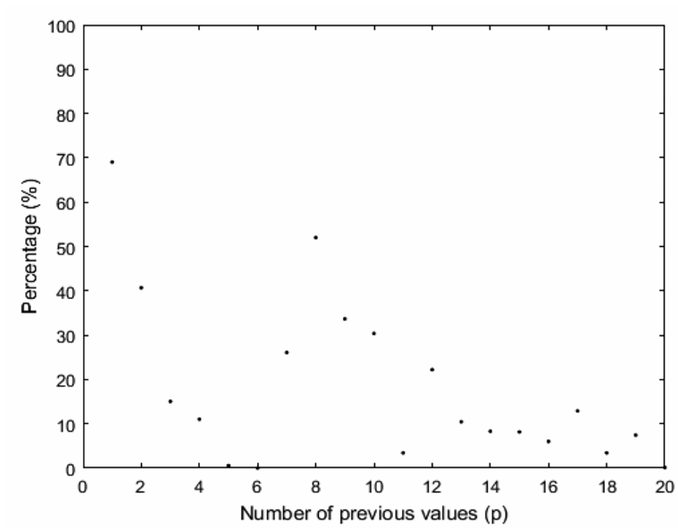
	NN-ARX		NN-ARMAX	
Station	%10 V	P	%10 V	P
A	68.4	2	69.1	1
B	67.2	2	68.9	1

In addition, Figures 4.5a and 4.5b plot $\%10V$ as a function of p for station A, for the NN-ARX and NN-ARMAX models, respectively, and Figures 4.6a and 4.6b show the same plots for station B. From Figures 4.5 and 4.6 it can be seen that the NN-ARX model exhibits a more consistent performance, with the values of $\%10V$ between 40% and 70% for all values of p.

At last, Figure 4.7a plots the relative errors of the NN-ARX model for $p = 2$ for the 718 validation samples of station A, and Figure 4.7b displays the same plot for the NN-ARMAX model for $p = 1$. Similarly, Figures 4.8a and 4.8b plot the relative errors of the NN-ARX model for $p = 2$ and for the NN-ARMAX model for $p = 1$, respectively, for the 698 validation samples corresponding to station B.

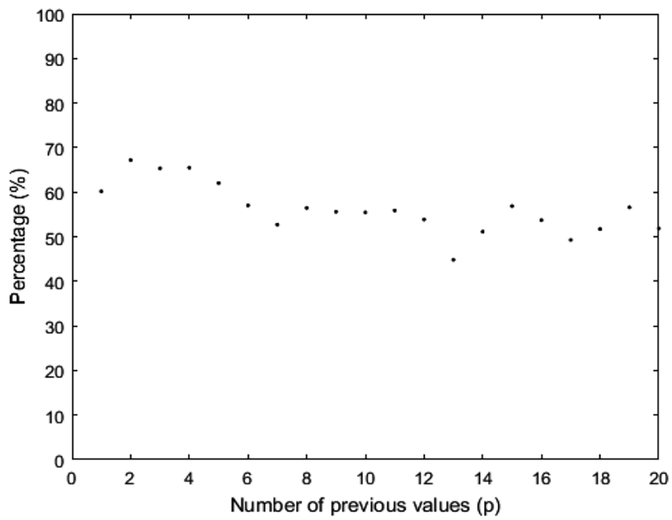


(a)

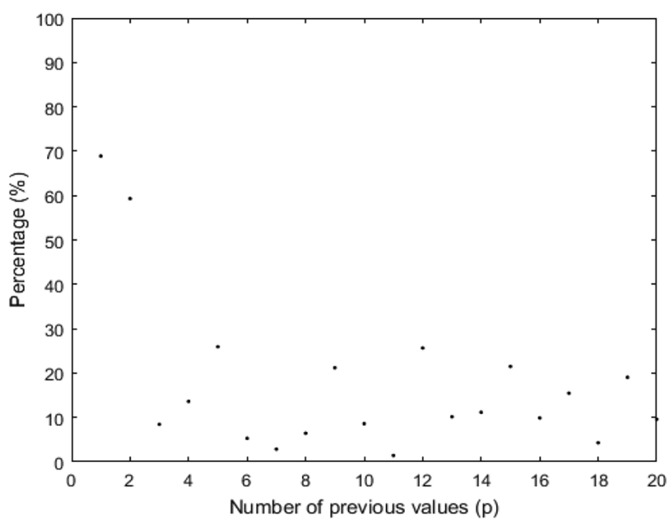


(b)

Figure 4.5: Values of %10 V as a function of p for station A: (a) NN_ARX model and (b) NN-ARMAX model.

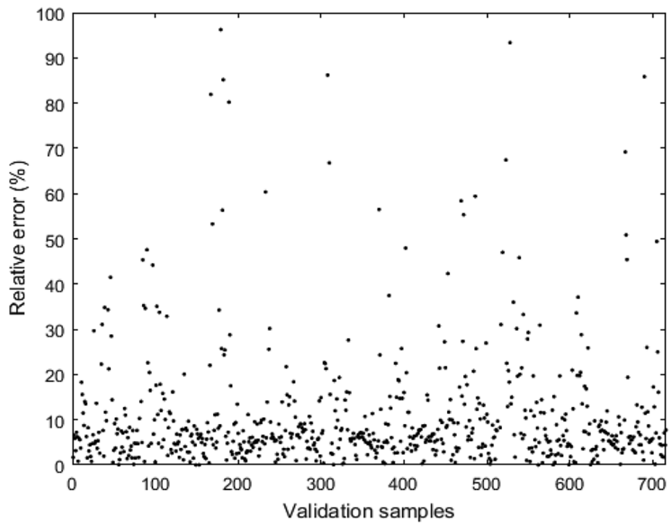


(a)

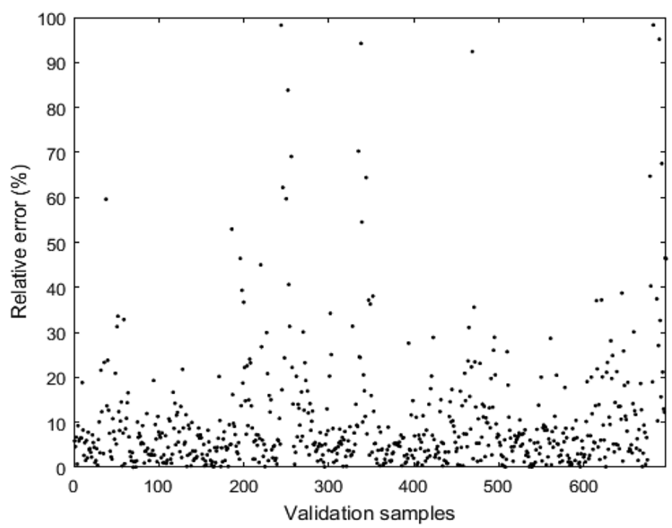


(b)

Figure 4.6: Values of %10 V as a function of p for station B: (a) NN_ARX model and (b) NN-ARMAX model.

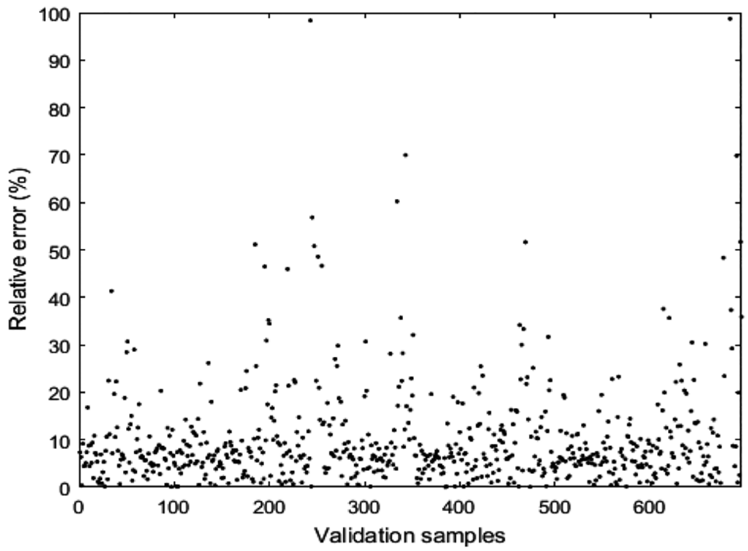


(a)

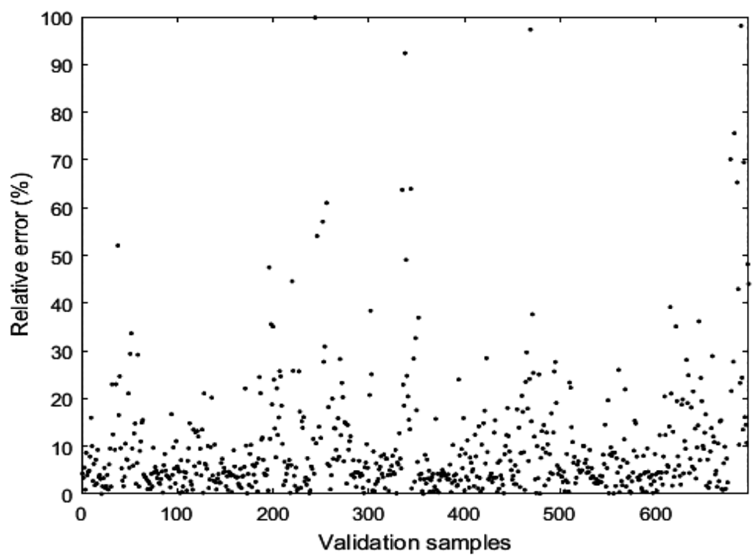


(b)

Figure 4.7: Relative errors on the validation samples for station A:(a) NN-ARX model for $p = 2$ and (b) NN-ARMAX model for $p = 1$.



(a)



(b)

Figure 4.8: Relative errors on the validation samples for station B: (a) NN-ARX model for $p = 2$ and (b) NN-ARMAX model for $p = 1$.

CONCLUSIONS

This work proposed the use of advanced time-varying approaches for representing the dynamics of the channel in wireless networks.

In the first case study, a linear time-varying Kalman innovation model was identified for describing RSSI as a function of the speed of the link, using 1561 samples measured in an indoor multipath wireless channel. The performance of the model was tested for orders varying between 1 and 20, and it showed to be (i) accurate, since for the best case (order 1) almost 100% of the samples exhibited a relative error less than 15%, and (ii) robust, because it yielded more than 95% of precision for orders less or equal to 16.

Then, the second case study considered testing the suitability of using linear time-varying Kalman innovation models of the RSSI, for secret key generation in the physical layer of a multipath wireless channel between two communicating parties. In particular, the Kalman models are identified for both users using a block of RSSI measurements, to describe the dynamics of the channel between them. After the models converged, the innovation errors (residuals) exhibit a significant potential for secret key generation because of their significant randomness content. Furthermore, the proposed approach outperformed other reported approaches for secret key generation, by yielding values of maximum channel capacity an order of magnitude greater for different signal-to-noise ratios.

At last, the third case study included the identification of a neural network-based autoregressive moving average with exogenous inputs (NN-ARMAX) model and a neural network-based autoregressive with exogenous inputs (NN-ARX) model, for describing traffic in a 4G-LTE network. According to the results obtained for different orders, both models exhibited a similar performance for the best case, achieving a relative error less than 10% in around 70% of the validation samples. It was also found that the NN-ARX model showed to be more consistent, but the NN-ARMAX model has the advantage that it can be converted to a linear time-varying Kalman innovation model, and thus can be used for the implementation of advanced schemes for controlling the operation of the network.

Additional features of the advanced modeling approaches include simplicity of implementation, high computational efficiency and small computational cost and excellent performance in modeling time-varying systems. With respect to the Kalman innovation modeling approach, it does not require to estimate the properties of the noise, in contrast with the standard Kalman model.

Since the advanced time-varying modeling approaches exhibited an outstanding performance, further research work will be first directed towards the extension of such approaches to identify models with multiple inputs and multiple outputs of wireless networks. Specifically, the inputs may comprise different operating parameters of the network, while the outputs may include different criteria that quantify QoE or that correspond to key performance indicators. Then, a second research topic will be related to the implementation of schemes to determine optimal operation schedules of wireless networks, based on the models identified.

Bibliography

[1] Junqing Zhang, Alan Marshall, Roger Woods, and Trung Q Duong. Secure key generation from ofdm subcarriers' channel responses. In *2014 IEEE Globecom Workshops (GC Wkshps)*, pages 1302–1307. IEEE, 2014.

[2] Junqing Zhang, Alan Marshall, Roger Woods, and Trung Q Duong. Efficient key generation by exploiting randomness from channel responses of individual ofdm subcarriers. *IEEE Transactions on Communications*, 64(6):2578–2588, 2016.

[3] Joao Barros and Miguel RD Rodrigues. Secrecy capacity of wireless channels. In *2006 IEEE international symposium on information theory*, pages 356–360. IEEE, 2006.

[4] Junqing Zhang, Biao He, Trung Q Duong, and Roger Woods. On the key generation from correlated wireless channels. *IEEE Communications Letters*, 21(4):961–964, 2017.

[5] Michael McGuire. Channel estimation for secret key generation. In *2014 IEEE 28th International Conference on Advanced Information Networking and Applications*, pages 490–496. IEEE, 2014.

[6] Rupul Safaya. *A Multipath Channel Estimation Algorithm using a Kalman filter*. PhD thesis, Illinois Institute of Technology, Chicago, IL, 1997.

[7] Zakia Jellali and Leïla Najjar Atallah. Fast fading channel estimation by kalman filtering and cir support tracking. *IEEE Transactions on Broadcasting*, 63(4):635–643, 2017.

[8] Stephen G Larew and David J Love. Adaptive beam tracking with the unscented kalman filter for millimeter wave communication. *arXiv preprint arXiv:1804.08640*, 2018.

[9] Alireza Movahedian and Michael McGuire. Estimation of fast-fading channels for turbo receivers with high-order modulation. *IEEE transactions on Vehicular Technology*, 62(2):667–678, 2012.

[10] Sandra Almeida, José Queijo, and Luis M Correia. Spatial and temporal traffic distribution models for gsm. In *Gateway to 21st Century Communications Village. VTC 1999-Fall. IEEE VTS 50th Vehicular Technology Conference (Cat. No. 99CH36324), Amsterdam, ND*, volume 1, pages 131–135. IEEE, 1999.

[11] Ana Margarida Pina Simões. *Temporal Modelling of Mobile Data Traffic Applications for Network Optimisation*. PhD thesis, ITB, Lisboa, Portugal, 2017.

[12] Chaoyun Zhang, Paul Patras, and Hamed Haddadi. Deep learning in mobile and wireless networking: A survey. *IEEE Communications Surveys & Tutorials*, 21(3):2224–2287, 2019.

[13] Demetrios Zeinalipour Yazti and Shonali Krishnaswamy. Mobile big data analytics: research, practice, and opportunities. In *2014 IEEE 15th International Conference on Mobile Data Management*, volume 1, pages 1–2. IEEE, 2014.

[14] Matías Toril, Salvador Pedraza, Ricardo Ferrer, and Volker Wille. Optimization of handover margins in gsm/gprs networks. In *The 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring, Jeju, South Korea*, volume 1, pages 150–154. IEEE, 2003.

[15] S Tang and R Tafazolli. Scalable resource allocation algorithm for gprs. In *The 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring, Jeju, South Korea*, volume 1, pages 165–170. IEEE, 2003.

[16] Wei Li, Zhiyong Feng, Qian Li, Vanbien Le, and T Aaron Gulliver. Dynamic spectrum management for wcdma and dvb heterogeneous systems. In *2010 IEEE Wireless Communication and Networking Conference*, volume 10, pages 1582–1593, 2010.

[17] Sunav Choudhary, Shaunak Mishra, Nachiket Desai, N Swathi Priya, Dhaval Chudasama, and RV Rajakumar. A fair cognitive channel allocation method for cellular networks. In *2009 Second International Workshop on Cognitive Radio and Advanced Spectrum Management, Aalborg, Denmark*, pages 138–142. IEEE, 2009.

[18] Byungchan Ahn, Hyunsoo Yoon, and Jung Wan Cho. A design of macro-micro cdma cellular overlays in the existing big urban areas. *IEEE journal on selected areas in communications*, 19(10):2094–2104, 2001.

[19] Angelo Furno, Marco Fiore, Razvan Stanica, Cezary Ziemlicki, and Zbigniew Smoreda. A tale of ten cities: Characterizing signatures of mobile traffic in urban areas. *IEEE Transactions on Mobile Computing*, 16(10):2682–2696, 2016.

[20] Byungchan Ahn, Hyunsoo Yoon, and Jung-Wan Cho. Joint deployment of macrocells and microcells over urban areas with spatially non-uniform traffic distributions. In *Vehicular Technology Conference Fall 2000. IEEE VTS Fall VTC2000. 52nd Vehicular Technology Conference (Cat. No. 00CH37152), Boston, MA*, volume 6, pages 2634–2641. IEEE, 2000.

[21] Yosef Abera and Dereje Hailemariam. Spatio-temporal mobile data traffic modeling using fourier transform techniques. In *2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, South Korea*, pages 20–24. IEEE, 2018.

[22] Zhenglei Yi, Xin Dong, Xing Zhang, and Wenbo Wang. Spatial traffic prediction for wireless cellular system based on base stations social network. In *2016 Annual IEEE Systems Conference (SysCon), Orlando, FL*, pages 1–5. IEEE, 2016.

[23] Jiandong Li, Hua Shi, Honghao Ju, and Jie Zheng. Queue-aware resource allocation scheme in hybrid macrocell-femtocell networks. In *2013 IEEE 78th Vehicular Technology Conference (VTC Fall), Las Vegas, NV*, pages 1–5. IEEE, 2013.

[24] Seok-Yee Tang, Shyamalie Thilakawardana, Rahim Tafazolli, and Yi Qian. Performance analysis of predictive scalable resource allocation for integrated wireless networks. In *2005 International Conference on Wireless Networks, Communications and Mobile Computing, Maui, HI*, volume 1, pages 745–750. IEEE, 2005.

[25] Sebastian Troia. *Machine learning-based traffic prediction and pattern extraction for dynamic optical routing in SDN mobile metro networks*. PhD thesis, Politecnico di Milano, Milano, Italy, 2016.

[26] Pedro Torres, Paulo Marques, Hugo Marques, Rogério Dionísio, Tiago Alves, Luis Pereira, and Jorge Ribeiro. Data analytics for forecasting cell congestion on lte networks. In *2017 Network Traffic Measurement and Analysis Conference (TMA), Dublin, Ireland*, pages 1–6. IEEE, 2017.

[27] Araya Eamrurksiri. *Applying Machine Learning to LTE/5G Performance Trend Analysis*. PhD thesis, Linköping University, Stockholm, Sweden, 2019.

[28] Emil Bergner. *Unsupervised learning of traffic patterns in self-optimizing 4th generation mobile networks*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2012.

[29] Luong Vy Le, Bao-Shuh Lin, Li-Ping Tung, and Sinh Do. Enhanced handover clustering and forecasting models based on machine learning and big data. *Transactions on Machine Learning and Artificial Intelligence*, 6(5):43–43, 2018.

[30] Luong-Vy Le, Do Sinh, Li-Ping Tung, and Bao-Shuh Paul Lin. A practical model for traffic forecasting based on big data, machine-learning, and network kpis. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV*, pages 1–4. IEEE, 2018.

[31] Hadi Jamali Rad, Toon Van Waterschoot, and Geert Leus. Cooperative localization using efficient kalman filtering for mobile wireless sensor networks. In *2011 19th European Signal Processing Conference, Barcelona, Spain*, pages 1–4. IEEE, 2011.

[32] James Haught, Kenneth Hopkinson, Nathan Stuckey, Michael Dop, and Alexander Stirling. A kalman filter-based prediction system for better network context-awareness. In *Proceedings of the 2010 Winter Simulation Conference, Baltimore, MD*, pages 2927–2934. IEEE, 2010.

[33] Biljana Bojović, Elena Meshkova, Nicola Baldo, Janne Riihijärvi, and Marina Petrova. Machine learning-based dynamic frequency and bandwidth allocation in self-organized lte dense small cell deployments. *EURASIP Journal on Wireless Communications and Networking*, 2016(1):183, 2016.

[34] Xuewu Dai, Wuxiong Zhang, Jing Xu, John E Mitchell, and Yang Yang. Kalman interpolation filter for channel estimation of lte downlink in high-mobility environments. *EURASIP Journal on Wireless Communications and Networking*, 232(1):2–14, 2012.

[35] Djorwé Témoa, Anna Förster, and Serge Doka Yamigno. A reinforcement learning based intercell interference coordination in lte networks. *Future Internet*, 1, 2019.

[36] Zoraze Ali, Nicola Baldo, Josep Mangues-Bafalluy, and Lorenza Giupponi. Machine learning based handover management for improved qoe in lte. In *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium, Istanbul, Turkey*, pages 794–798. IEEE, 2016.

[37] Aasia Kashaf, Moazzam Islam Tiwana, Imran Usman, and Mohsin Islam Tiwana. Self-organizing inter-cell interference coordination in 4g and beyond networks using genetic algorithms. *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, 58(1):48–54, 2017.

[38] Sameh Musleh, Mahamod Ismail, and Rosdiadee Nordin. Load balancing models based on reinforcement learning for self-optimized macro-femto lte-advanced heterogeneous network. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9(1):47–54, 2017.

[39] Brad Kyoshi Donohoo. *Machine learning techniques for energy optimization in mobile embedded systems*. PhD thesis, Colorado State University, Libraries, 2012.

[40] Jing Wang, Jian Tang, Zhiyuan Xu, Yanzhi Wang, Guoliang Xue, Xing Zhang, and Dejun Yang. Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach. In *36th Annual IEEE International Conference on Computer Communications (INFOCOM), Atlanta, GA, USA*, pages 1–9. IEEE, 2017.

[41] Chaoyun Zhang and Paul Patras. Long-term mobile traffic forecasting using deep spatio-temporal neural networks. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, Los Angeles, CA*, pages 231–240, 2018.

[42] Chaoyun Zhang, Xi Ouyang, and Paul Patras. Zipnet-gan: Inferring fine-grained mobile traffic patterns via a generative adversarial neural network. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies, Seoul/Incheon, South Korea*, pages 363–375, 2017.

[43] Chih-Wei Huang, Chiu-Ti Chiang, and Qiuhui Li. A study of deep learning networks on mobile traffic forecasting. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC*, pages 1–6. IEEE, 2017.

[44] Chuanting Zhang, Haixia Zhang, Dongfeng Yuan, and Minggao Zhang. Citywide cellular traffic prediction based on densely connected convolutional neural networks. *IEEE Communications Letters*, 22(8):1656–1659, 2018.

[45] Shiva Navabi, Chenwei Wang, Ozgun Y Bursalioglu, and Haralabos Papadopoulos. Predicting wireless channel features using neural networks. In *2018 IEEE international conference on communications (ICC), Kansas City, MO*, pages 1–6. IEEE, 2018.

[46] Xu Wang, Zimu Zhou, Fu Xiao, Kai Xing, Zheng Yang, Yunhao Liu, and Chunyi Peng. Spatio-temporal analysis and prediction of cellular traffic in metropolis. *IEEE Transactions on Mobile Computing*, 18(9):2190–2202, 2018.

[47] Abdelkader Aljerme and Huaping Liu. Efficient kalman modeling of multipath wireless channel for secret key generation. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 586–590. IEEE, 2019.

[48] Abdelkader Aljerme and Huaping Liu. Time-varying kalman modeling of measured rssi in multipath wireless channels. In *Accepted in WASET international conference on computing & communications, June 2020*. IEEE, 2020.

[49] Siegmund M Redl, Matthias K Weber, and Malcolm W Oliphant. *An introduction to GSM*. Norwood, MA: Artech House Inc., 2005.

[50] Claude E Shannon. Communication theory of secrecy systems. *Bell system technical journal*, 28(4):656–715, 1949.

[51] Abdelkader Aljerme and Huaping Liu. A time-varying neural-network-based arx model of the traffic in mobile networks. In *Presented at the 15th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2020.

[52] Rahul Khanna, Huaping Liu, and Hsiao-Hwa Chen Chen. Self-organization of sensor networks using genetic algorithms. In *2006 IEEE International Conference on Communications (ICC'06)*, pages 3377–3382. IEEE, 2006.

[53] Rahul Khanna, Huaping Liu, and Hsiao-Hwa Chen Chen. Dynamic optimization of secure mobile sensor networks: a genetic algorithm. In *2007 IEEE International Conference on Communications (ICC'07)*. IEEE, 2007.

[54] Rahul Khanna and Huaping Liu. Control theoretic approach to intrusion detection using distributed hidden markov model. *IEEE Wireless Communications*, 15(4):24–33, 2008.

[55] Rahul Khanna and Huaping Liu. System approach to intrusion detection using hidden markov model. In *2008 IEEE International Conference on Communications (ICC'08)*. IEEE, 2008.

[56] Rahul Khanna, Huaping Liu, and Hsiao-Hwa Chen. System approach to intrusion detection using hidden markov model. In *2009 IEEE International Conference on Communications (ICC'09)*. IEEE, 2009.

[57] Yuexing Peng, George C Alexandropoulos, Peng Wang, Yonghui Li, and Dac-Binh Ha. Poster: Secret key generation from cfr for ofdm tdd systems over fading channels. In *9th International Conference on Communications and Networking in China*, pages 660–661. IEEE, 2014.

[58] Youssef El Hajj Shehadeh and Dieter Hogrefe. A survey on secret key generation mechanisms on the physical layer in wireless networks. *Security and Communication Networks*, 8(2):332–341, 2015.

[59] Hongbo Liu, Yang Wang, Jie Yang, and Yingying Chen. Fast and practical secret key extraction by exploiting channel response. In *The 32nd IEEE International Conference on Computer Communications, Turin, Italy*, pages 3048–3056. IEEE, 2013.

[60] Ahmed Badawy, Tarek Elfouly, Tamer Khattab, Amr Mohamed, and Mohsen Guizani. Unleashing the secure potential of the wireless physical layer: Secret key generation methods. *Physical Communication*, 19:1–10, 2016.

[61] Suhas Mathur, Wade Trappe, Narayan Mandayam, Chunxuan Ye, and Alex Reznik. Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 128–139, 2008.

[62] Jingjing Huang and Ting Jiang. Dynamic secret key generation exploiting ultra-wideband wireless channel characteristics. In *2015 IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, LA*, pages 1701–1706. IEEE, 2015.

[63] Hongbo Liu, Jie Yang, Yan Wang, and Yingying Chen. Collaborative secret key extraction leveraging received signal strength in mobile wireless networks. In *31st Annual IEEE International Conference on Computer Communications (IEEE INFOCOM 2012), Orlando, FL*, pages 927–935. IEEE, 2012.

[64] Yanpei Liu, Stark C Draper, and Akbar M Sayeed. Exploiting channel diversity in secret key generation from multipath fading randomness. *IEEE Transactions on information forensics and security*, 7(5):1484–1497, 2012.

[65] Alireza Movahedian and Michael McGuire. Estimation of fast-fading channels for turbo receivers with high-order modulation. *IEEE transactions on Vehicular Technology*, 62(2):667–678, 2012.

[66] Angelo Furno, Razvan Stanica, and Marco Fiore. A comparative evaluation of urban fabric detection techniques based on mobile traffic data. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Paris, France*, pages 689–696. IEEE, 2015.

[67] Kurt Hornik, Maxwell Stinchcombe, Halbert White, et al. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[68] Jose I Canelon, Leang S Shieh, Yongpeng Zhang, and Cajetan M Akujuobi. A new neural network-based approach for self-tuning control of nonlinear multi-input multi-output dynamic systems. In *Presented at American Control Conference, St. Louis, MI*, pages 3561–3566. IEEE, 2009.

[69] Nan-Ying Liang, Guang-Bin Huang, Paramasivan Saratchandran, and Narasimhan Sundararajan. A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on neural networks*, 17(6):1411–1423, 2006.

[70] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[71] Andrea Goldsmith. Wireless communication, 2012.

[72] John W Betz. *Engineering satellite-based navigation and timing: global navigation satellite systems, signals, and receivers*. Piscataway, NJ: IEEE Press Wiley, 2016.

[73] Laurene Fausett. *Fundamentals of neural networks: architectures, algorithms, and applications*. Prentice-Hall, Inc., 1994.

[74] Robert Hecht-Nielsen. Neurocomputing: picking the human brain. *IEEE spectrum*, 25(3):36–41, 1988.

[75] Simon Haykin. *Neural networks: a comprehensive foundation*. John Wiley & Sons, 2001.

[76] G Gybenko. Approximation by superposition of sigmoidal functions. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

[77] Roberto Battiti. First-and second-order methods for learning: between steepest descent and newton's method. *Neural computation*, 4(2):141–166, 1992.

[78] Guang-Bin Huang, Lei Chen, Chee Kheong Siew, et al. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Networks*, 17(4):879–892, 2006.

[79] Ljung Lennart. System identification: theory for the user. *PTR Prentice Hall, Upper Saddle River, NJ*, pages 1–14, 1999.

[80] Marcelo CM Teixeira and Stanislaw H Zak. Stabilizing controller design for uncertain nonlinear systems using fuzzy models. *IEEE Transactions on Fuzzy systems*, 7(2):133–142, 1999.

[81] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, and Elaine Barker. A statistical test suite for random and pseudorandom number generators for cryptographic applications. *NIST Special Publication*, 800:22, 2010.

[82] Chunxuan Ye, Suhas Mathur, Alex Reznik, Yogendra Shah, Wade Trappe, and Narayan B Mandayam. Information-theoretically secret key generation for fading wireless channels. *IEEE Transactions on Information Forensics and Security*, 5(2):240–254, 2010.

[83] Michael McGuire. Channel estimation for secret key generation. In *2014 IEEE 28th International Conference on Advanced Information Networking and Applications, Victoria, Canada*, pages 490–496. IEEE, 2014.

[84] Rupul Safaya. *A Multipath Channel Estimation Algorithm using a Kalman filter.* PhD thesis, Illinois Institute of Technology, Chicago, IL, 1997.

[85] Zakia Jellali and Leïla Najjar Atallah. Fast fading channel estimation by kalman filtering and cir support tracking. *IEEE Transactions on Broadcasting*, 63(4):635–643, 2017.

[86] Stephen G Larew and David J Love. Adaptive beam tracking with the unscented kalman filter for millimeter wave communication. *arXiv preprint arXiv:1804.08640*, 2018.

[87] DE Rumelhart, GE Hinton, and RJ Williams. Learning internal representations by error propagation. In *In Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, pages 318–362. Cambridge, MA: MIT Press, 1986.