NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

TIAGO NUNO BARROS SOUSA

Electrical and Computer Engineering

BICYCLES' MOBILITY PREDICTION

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon

March, 2022

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL
AND COMPUTER ENGINEERING

BICYCLES' MOBILITY PREDICTION

TIAGO NUNO BARROS SOUSA

Electrical and Computer Engineering

Adviser: Rodolfo Oliveira
Associate Professor with Habilitation, NOVA University Lisbon

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
March, 2022

Bicycles' Mobility Prediction

Acknowledgements

Primeiramente, é crucial agradecer à minha Mãe e ao meu Pai por toda a força que me deram ao longo destes anos. Sem eles, o fechar desta etapa não teria sido possível. De seguida agradecer ao meu irmão por todos os conselhos e ensinamentos, e à minha irmã por todo o companheirismo e suporte. À minha restante família, também agradecer todo o apoio e conselhos que me foram dando ao longo da minha vida. De forma geral, e a todos os referenciados anteriormente, é bom ter uma família que deposita confiança em nós e que nos faz voar sem nos prender as asas. Que estão sempre lá nos bons e maus momentos. Obrigado.

Seguidamente, quero agradecer à que foi a minha segunda casa durante cinco anos - Faculdade de Ciências e Tecnologia da Universidade de Lisboa. Ao meu orientaddor, Professor Rodolfo Oliveira, por todas as ajudas, assim como toda a documentação e tempo disponibilizado ao longo da realização desta dissertação.

Quero agradecer a todos os meus amigos da FCT e a todas as pessoas que se cruzaram comigo nos corredores desta faculdade e que de alguma forma, marcaram o meu percurso académico. A esses amigos, agradecer toda a cooperação e companhia ao longo deste tempo.

Um grande Obrigado aos meus amigos de Setúbal, amigos de vários anos. Agradecer por todas as aventuras, amizade e memórias que vou levar para a vida.

Por fim, mas não menos importante, um agradecimento especial à minha namorada pelo apoio e motivação incansáveis, sendo um pilar fundamental nestes últimos meses.

"You have to know the past to understand the present." (Carl Sagan)

Abstract

The growth in mobile wireless communication requires sharp solutions in handling mobility problems that encompass poor handover management, interference in access points, excessive load in macrocells, and other relevant mobility issues. With the deployment of small cell networks in 5G mobile systems the problems mentioned intensify thus, mobility prediction schemes arise to surpass and mitigate these issues. Predicting mobility is not a trivial task due to the vastness of different variables that characterize a mobility route translating into unpredictability and randomness. Therefore, the task of this work is to overcome these challenges by building a solid mobility prediction architecture that can analyze big data and find patterns in the mobility aspect to ultimately perform reliable predictions. The models introduced in this dissertation are two deep learning schemes based on an Artificial Neural Network (ANN) architecture and a LSTM Long-Short Term Memory (LSTM) architecture. The prediction was made in two levels: Short-term prediction and Long-term prediction. We verified that in the short-term domain both models performed equivalently with successful results. However, in long-term prediction, the LSTM model surpassed the ANN model. Consequently, the LSTM approach constitutes the stronger model in all prediction aspects. Implementing this model in cellular networks is an important asset in optimizing processes such as routing and caching as the cellular networks can allocate the necessary resources to provide a better user experience. With this optimization impact and with the emergence of the Internet of Things (IoT), the prediction model can support and improve the development of smart applications related to our daily mobility routine.

Keywords: Mobility prediction, Machine Learning, Deep learning, Performance evaluation.

Resumo

O crescimento da comunicação móvel sem fios exige soluções precisas para lidar com problemas de mobilidade que englobam uma gestão pobre de handover, interferência em pontos de acesso, carga excessiva em macrocélulas e outros problemas relevantes ao aspeto da mobilidade. Com a implantação de redes de pequenas células no sistema móvel 5G, os problemas mencionados intensificam-se. Desta forma, são necessários esquemas de previsão de mobilidade para superar e mitigar esses problemas. Prever a mobilidade não é uma tarefa trivial devido à imensidão de diferentes variáveis que caracterizam uma rota de mobilidade, traduzindo-se em grandes dimensões de imprevisibilidade e aleatoriedade. Portanto, a tarefa deste trabalho é superar esses desafios construindo uma arquitetura sólida de estimação de mobilidade, que possa analisar um grande fluxo de dados e encontrar padrões para, em última análise, realizar previsões credíveis e assertivas. Os modelos apresentados nesta dissertação são dois esquemas de deep learning baseados em uma arquitetura de RNA (Rede Neuronal) e uma arquitetura LSTM (Long-Short Term Memory). A previsão foi feita em dois níveis: previsão de curto prazo e previsão de longo prazo. Verificámos que no curto prazo ambos os modelos tiveram um desempenho equivalente com resultados bem sucedidos. No entanto, na previsão de longo prazo, o modelo LSTM superou o modelo ANN. Consequentemente, a abordagem LSTM constitui o modelo mais forte em todos os aspectos de previsão. A implementação deste modelo, em redes celulares, é uma medida importante na otimização de processos como, routing ou caching, proporcionando uma melhor experiência wireless ao utilizador. Com este impacto de otimização e com o surgimento da Internet of Things (IoT), o modelo de previsão pode apoiar e melhorar o desenvolvimento de aplicações inteligentes relacionadas com a nossa rotina diária de mobilidade.

Palavras-chave: Previsão de mobilidade, Aprendizagem de máquina, Aprendizagem profunda, Avaliação de desempenho.

Contents

List of Figures

List of Tables

Acronyms

3GPP	3rd Generation Partnership Project
4G	4th Generation
5G	5th Generation
AI	Artificial Intelligence
ANN	Artificial Neural Network
AP	Access Point
BS	Base Station
CDF	Cumulative Distribution Function
CKF	Constant Kalman Filter
CMS	Current Movement State
CMSA	Current Movement State Based Approaches
CoMP	Coordinated Multi Point
DFTS	Discrete Fourier Transform Spread
eMBB	Enhanced Mobile Broadband
eNB	Evolved Node B
EOF	End-Of-File
GMM	Gaussian Mixture Model
GPS	Global Positioning System
HDA	Hybrid Approaches
HetNets	Heterogeneous Networks
HMM	Hidden Markov Model

HMP	Historical Movement Pattern
HMPA	Historical Movement Pattern Based Approaches
IMT-A	International Mobile Telecommunications-Advanced
IoT	Internet of Things
ITS	Intelligent Transportation Systems
ITU	International Telecommunication Union
JSTM	Joint Time Series Model
LSTM	Long Short-Term Memory
LTE	Long Term Evolution
LTE-A	Long Term Evolution-Advanced
MC	Markov Chain
MIMO	Multiple Input Multiple Output
ML	Machine Learning
MLP-BP	Multilayer Perceptron Backpropagation Neural Network
MLSTM	Multiple Long Short-Term Memory
MME	Mobility Management Entity
mMTC	Massive Machine-Type Communications
MT	Mobile Terminal
OFDM	Orthogonal Frequency-Division Multiplexing
PP	Prediction Performance
PPN	Polynomial Perceptron Network
QoS	Quality of Service
ReLU	Rectified Linear Activation Function
RNN	Recurrent Neural Network
RSRP	Reference Signals Received Power
SC-FDE	Single-Carrier Frequency Domain Equalization
SINR	Signal-to-Interference-plus-Noise Ratio
SLAM	Simultaneous Localization and Mapping
ST-LSTM	Spatio-Temporal-Long-Short-Term Memory
SVM	Support Vector Machine
SVR	Support Vector Regression

TA	Tracking Area
TP	Transmission Point
UDN	Ultra-Dense Networks
UE	User Equipment
UMTS	Universal Mobile Telecommunication System
URLLC	Ultra-reliable and Low Latency Communications
USRP	Universal Software Radio Peripheral
UTRA	UMTS Terrestrial Radio Access Network
V2V	Vehicle-To-Vehicle

1

Introduction

Mobile wireless communication has experienced a sharp growth over the recent years, mainly due to the breakthrough of affordable mobile devices, as well as the innovative deployment of small cell networks that will be a trademark of the 5th Generation mobile system. The increase in traffic demand and the network densification will inevitably stimulate mobility problems, principally in a poor handover management (e.g., unacceptable delay, call dropping/blocking events) if there are no suitable mobility prediction schemes. The existing models, such as Markov Chain (MC), Hidden Markov Model (HMM), and Bayesian Networks, are stochastic processes based on probabilistic transitions. Under the scope of probability, the trend of Artifical Neural Networks (ANNs) emerge as a more reliable method for prediction accuracy insofar as with the right metrics and inputs, the network will be capable of learning and generate the best output to a given location. Presuming that we can estimate the user's future position and trajectory, it enables an efficient resource management, where the Transmission Points (TPs) can allocate the necessary resources to ensure the Quality of Service (QoS). However, predicting a future user location through the "lens" of the network constitutes a hard challenge because of the immeasurable variables representing the unpredictability and randomness of a route.

1.1 Motivation

Over the years, mobility prediction has triggered a rapid evolution in search of solutions related to vehicular trajectory. With the appearance of the Internet of Things (IoT), a boost in connected vehicles and data-driven intelligent transportation systems (ITS) has started. The huge amount of data originates a degradation of the QoS, where data loss and unreliability are a matter of concern. The development of efficient and personalized prediction algorithms based on mobility patterns, allows a deeper understanding of the construction and design of safer and efficient systems. The ability of a system to predict a trajectory will enable a panoply of interactions between the user and the mobile center, which allows a more sophisticated and easier way of communicating and also fulfilling the user's requirements. Similar to the study of vehicular mobility, human mobility

focused on pedestrians' activities constitutes an important subject. A wide range of activities such as, walking, biking, and other movement-related activities represent an indispensable fraction of our daily lives. Many challenges must be considered when performing mobility prediction in pedestrians. If the scenario represents an open-air space where users perform their daily workout routine, a simple analysis on the user's trajectory data will solve the problem. In a more unpredictable area such as a whole city, other approaches are required in order to predict a random displacement accurately. The latter defines the main motivation of this dissertation, where solutions for these types of challenges regarding the user's dynamic behaviors and the environment will be addressed.

1.2 Objectives and Contributions

This section describes the aimed objectives and the contributions reached in this dissertation.

Objectives:

The main goal of this thesis is to design and build a mobility prediction scheme based on bicycles' mobility. To correctly complete this work, the procedure should be as follow:

1. Gather knowledge in how to decide on a suitable area to perform mobility prediction. Afterwards, we need to choose an appropriate dataset that contains relevant data for the implementation of the model.
2. Evaluation of the dataset, by pre-processing the raw data in order to improve the efficiency of the algorithm.
3. Designing of an individual trajectory prediction algorithm based on Machine Learning (ML) techniques, i.e. Artificial Neural Networks (ANN), Long Short-Term Memory (LSTM).
4. Implementation of the fully connected architecture, and definition of adequate metrics to accurately characterize the model.
5. Performance evaluation of the mobility prediction methodologies

Contributions:

- Evaluation of the dataset features and design of a bi-dimensional grid that incorporates a whole city area.
- Construction of two efficient and adaptive mobility prediction algorithms, that provide reliable predictions to any given environment.
- Performance assessment of both prediction models.

1.3 Dissertation Structure

The report structure is organized as follows: Chapter 2 holds a literature review about related work. An overview of different mobility prediction schemes will be presented as well as their impact on optimizing the network. Chapter 3 presents a definition of the data used focusing on the analytic characteristics that enable the construction of the architectures. Chapter 4 describes the results of both ANN and LSTM deep learning techniques and expresses a comparative evaluation of the models. Finally, Chapter 5 contains the final considerations and closes the thesis with the final remarks and improvements aspects of the performed work.

2

Related Work

2.1 Long Term Evolution

Due to the technological breakthrough of mobile networks and devices, the world is currently facing a rise in traffic volume, described by the number of users and the diversification of the services provided by mobile devices. Consequently, there was the necessity to provide better throughputs, larger user capacity and greater spectral efficiency in mobile networks. Universal Mobile Telecommunication System (UMTS) LTE, also known as the 4th Generation (4G) of mobile telecommunications, is a standard for wireless broadband technology for mobile devices and data terminals, designated by the 3rd Generation Partnership Project (3GPP) to cope with this demand where its main objective is to support internet-based services on mobile devices. 3GPP defined the specifications for UMTS LTE named UMTS Terrestrial Radio Access (UTRA) and UMTS Terrestrial Radio Access Network in Release 8 [1]. Consisting of a wide band channel it enabled peak data rates that exceeded 300 Mb/s on the downlink and 75 Mb/s on the uplink. It was one of the first releases to be commercialized and it is principally installed in a macro/micro cell layout and uses OFDM as the downlink multiple access scheme and Discrete Fourier Transform Spread (DFTS)-OFDM, also known as SC-FDE, as the uplink multiple access schemes [2].

2.1.1 LTE-Advanced

LTE-A, also known as 3GPP Release 10, represents a significant improvement of LTE Release 8, enabling a better user experience. It was designed to meet the International Mobile Telecommunications-Advanced (IMT-A) requirements defined by the International Telecommunication Union (ITU) [2]. Carrier aggregation, downlink and uplink spatial multiplexing and Coordinated MultiPoint (CoMP) transmission and reception portray the techniques used in this scheme.

LTE-A is an essential step towards the transition from the 4G to the 5G wireless system. 5G networks will handle the expected traffic volume explosion and the new specifications through a conjunction of evolved existing technologies and new radio concepts [3].

2.1.2 5G

The existing technologies that contemplate 5G are Heterogeneous networks (HetNets), massive multiple inputs multiple outputs (MIMO), mmWave communication, CoMP and, the trend of 5G networks, ultra-dense networks (UDN), which is dense deployment of small cells [4]. The aforementioned represent the approach to the three main usage scenarios [5][6]:

- *Enhanced mobile broadband (eMBB)*: High throughput, wide-area coverage and hotspots for flawless user experience.
- *Ultra-reliable and low latency communications (URLLC)*: High throughput, low latency and high availability.
- *Massive machine-type communications (mMTC)*: Large coverage, relation low-cost devices/long battery life enhanced.

2.2 Mobility Management in cellular networks

As depicted in 2.1, the mobility management in LTE is handled by the mobility management entity (MME). According to [7], MME is linked to numerous of Evolved Node Base stations (eNB). The radio coverage of the eNBs is known as a cell, and each cell has a unique cell ID. Fig.1 illustrates groups of cells, that are known as tracking areas (TAs). The assignment of each cell to his TA is done upon the network planning phase and is determined in the operation phase. TAs are also represented by a unique ID (e.g. TA1), where this identification is broadcasted in the coverage area. TAs are grouped into lists (TALs), that are customized for each user equipment (UE). MME supplies the UE with the TAL that contains the TA in which it occupies [6]. Due to the implementation of UDN in 5G, the decrease of the cell coverage radius will amplify the mobility issues, concretely [7]:

1. *Handover Management*: Handover is a process in which data transmission is transferred from one base station to another without loosing its connectivity. In the networks aforementioned, frequent handover and the consequent high switching latency will degrade the QoS.
2. *Access Points Grouping*: In LTE communication systems, the choice of the next accessing cell is done regarding the Reference Signals Received Power (RSRP) and cell load. In 5G, due to the dense deployment, the cellular networks will simply saturate because of the frequent execution of update procedures.
3. *Resource Reservation*: As in LTE, 5G will experience a sharp growth of UE. Consequently, radio bandwidth resource shortage will be an imminent problem.

4. *Interference Coordination*: Interference is effect of Access Points (APs) that share the same resource blocks, this obstructs the implement of massive MIMO and other scenarios.

5. *Load Balancing*: HetNets are a promising approach in alleviating the load of a "busy"macrocell. These networks can redistribute parts of the traffic to small cells to prevent the network from congesting and collision problems.

Mobility prediction arises as an interesting solver to these issues. With prior knowledge of user's data (eg. handover history, speed), we can map the range of the TPs and allocate the necessary resources for the base stations, as well as, reduce the execution of update procedures, coordinate the interference and diminish the load by predicting fluctuations in the traffic load.

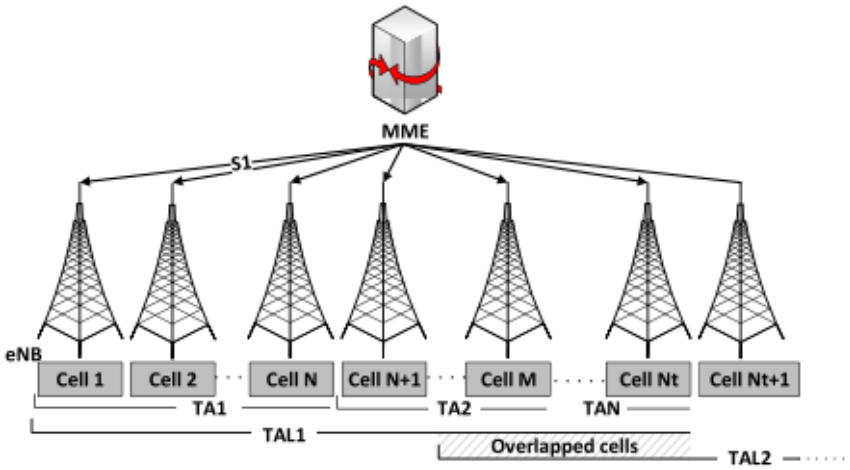

Figure 2.1: LTE mobility management architecture (taken from [7]).

2.3 Mobility Prediction Methods

In simple terms, mobility prediction is the ability to predict a user's path nonetheless its randomness and unpredictability. Hongtao and Lingcheng [7] raised a question regarding the predictability of the user's behavioral characteristics through multiple observations. The answer to the latter comes in the concept of entropy. Entropy is the average level of uncertainty, or more accurately, the level of information that will dictate the possibles outcomes of a random variable. Formally, it is defined by

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_b P(x_i). \tag{2.1}$$

In(2.1), $\sum$ denotes the sum over the variable's possible values and log denotes the logarithm, where the choice of base b differs between base 2 and base 10 depending

on the application. By measuring the entropy of user's individual trajectory [8] found that there is a potential 93% average predictability in user mobility. What indicates this value? It indicates that mobility is not purely random and it can be analysed through prediction techniques that enable pattern recognition by monitoring a user movement and history. In the next subsections, we are going to detail these prediction techniques such as: Markov chain, HMM, Bayesian Network, Data Mining and ANNs.

2.3.1 Markov Chain

A Markov chain is a stochastic model that characterizes a sequence of possible events. This method operates in a "chain reaction"system, meaning, the probability of each event depends directly on the outcome of the present event ("memorylessness"). If we approach this method in terms of mobility, the present event consists of our present state/location (present cell) and the future state/location represents our destination (next cell). This decision is not purely random since our "source"path was once our "destination"path, thus we can establish probabilities to each transition to obtain the most probable future path. An example of a cellular network scenario is presented in fig.2. The three cells build our "chain"and the transition probabilities (a...i) are defined. For instance, if a user is attached to cell "B"it has 3 possible transitions: Stay at the current cell, move to cell "i"or move to cell "a"with probabilities of "e", "f"and "d"respectively. As illustrated in 2.2, P represents the transition probability matrix, where the sum of each row/column is equal to 1. The next transition probability from the present cell to the future cell is calculated as:

$$P_n = [P_{n-1}] * [P]. \tag{2.2}$$

where, n denotes the transition to the next cell and n-1 the current transition probability matrix. The main issue of the Markov Chain is that does not analyze the previous states only the current state. Gambs *et al.* [9] stated that this property would decrease the prediction performance. To solve this problem, the author proposed a novel Markov chain named n-MMC which includes the n previous states. However, this extended version causes a major problem which translates in the complexity of the transition probability matrix thus being inappropriate to a large number of BSs networks. The success of this prediction algorithm comes in acquiring the transition probability matrix and balancing it with the complexity of the network, meaning that this method is not suitable for large networks. Moreover, studies found that the time required to collect the data is directly correlated with the performance of the algorithm. Therefore, the "key"in using this scheme relies on selecting an appropriate state space and pre-processing the data accordingly.

2.3.2 Hidden Markov Model

In a Markov Chain, the only parameters that are considered are the transition probabilities of the observable states (eg., the three cells in fig.2). The Hidden Markov Model

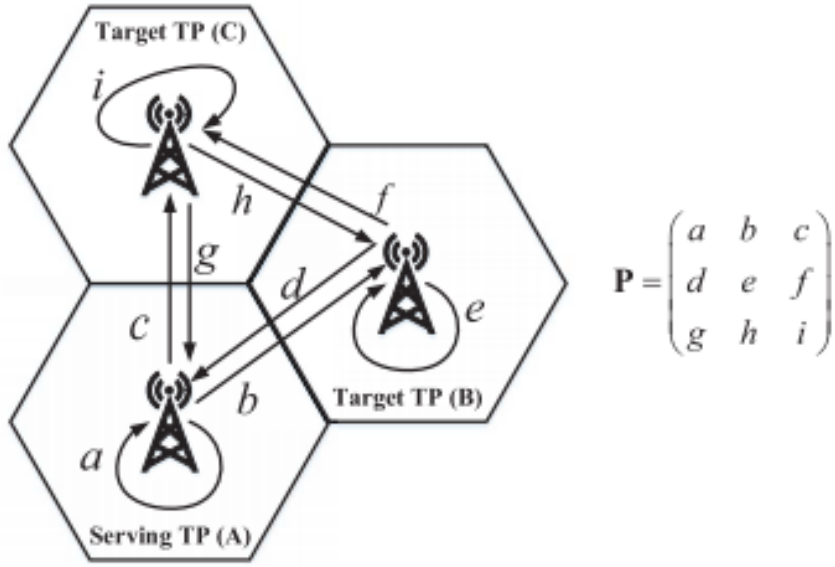

Figure 2.2: Application of a markov chain in a cellular scenario (taken from [7]).

differs from this process as it considers another set of states named unobservable or hidden. This double stochastic feature enables the observable states to obtain extra information through the hidden states. Qiao *et al.*[10] proposed a HMM to solve the challenge of trajectory prediction in transportation networks. The observable states represent cells and the hidden states trajectory segments. The Forward algorithm is used to solve the trajectory evaluation problem and to discover the sequence of hidden states the author applied Viterbi algorithm. The interesting fact of this model is the ability to evaluate objects with dynamic features (such as speed) being an important work in studying real-world scenarios. A similar approach was presented in [11], where the hidden states represent a sequence of visited cells. HMM can also be used in the optimization of power consumption and QoS [12] presented an intelligent location-awareness access point selection algorithm based on HMM. The outcome showed that the number of connections to high signal level AP increased and the number of connections to low signal level AP decreased. In relation to the Markov Chain, HMM is a more enhanced prediction algorithm as it reduces the lost information and it can constantly learn and update its output. The main flaw of this scheme is the "price"of the hidden states which results in an "expensive"computational task. Moreover, with the advent of UDN the computation complexity of an HMM approach would increase exponentially.

2.3.3 Bayesian Network

A Bayesian network is a direct acyclic graph that utilizes Bayesian inference for probability computations. Each edge corresponds to a conditional dependency and each node to a unique random variable. Following these associations, we can conduct inference on the random variables through the use of factors. Bayesian network has been widely

used because of its ability in solving uncertainty. For example, Liu *et al.*[13] proposed an approach in predicting a moving object's future path under uncertainty. As illustrated in fig.3, the Bayesian network was constructed in the following sequence:

1. Space-partitioning schemes;
2. Popular regions extraction;
3. Transformation of trajectory sequence and region sequence;
4. Frequent sequential pattern mining;
5. Bayesian network construction.

Similar to Markov chain and HMM, Bayesian network fails in being efficient in large networks, as exact inference is computationally expensive. Hence, the complete procedure in constructing a Bayesian network has to be shortened. Another issue is the deployment of small cells which will make the partition of the space challenging (first step).

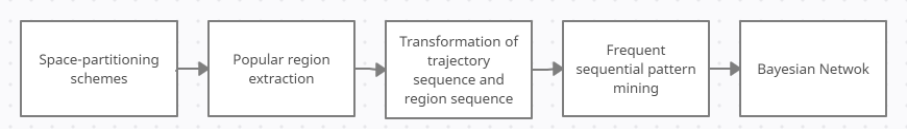

Figure 2.3: Design of Bayesian network.

2.3.4 Data Mining

Data mining is a product of three interlaced scientific domains: statistics, artificial intelligence, and machine learning. By definition, it is the process of "digging"(mining) through data to discover patterns, hidden connections and predict future trends. Data mining can be used based on different knowledge such as: Road topology information, user behavioral information, and movement parameters. A more detailed analysis can be found in [4].

Road Topology Information

The integration of road topology information constitutes an asset in improving the accuracy of prediction algorithms, as mobile terminals (MTs) travel on roads with rough characteristics that interfere with radio wave propagation. In [14] the author proposed a mobility prediction scheme that combined spatial conceptual maps and user's knowledge to predict traveling trajectory and destination. The results showed that this scheme maintains the same degree of accuracy independently of the period time, reason being, of the customized user strategy that does not need to obtain data of previously visited locations.

Road topology can be acquired through Global Positioning System (GPS) or Signal-to-Interference-plus-Noise-Ratio (SINR). However, the map can be difficult to obtain, and with modifications on the road the performance of the algorithm can decrease. Implementation of autonomous navigation techniques such as Simultaneous Localization and Mapping (SLAM) would be an interesting approach since the mapping of an unknown environment can be constructed and updated in real-time.

User Behavioral Information

Characterizing user's behaviors based on features, such as location, time of day, group pattern, and other relevant emerges as an interesting and intuitive approach in the design of prediction models. Duong and Tran [15] proposed a mobility scheme based on clustering and sequential pattern mining. Based on the premise that global scale users behave as groups depending on social and geographic constraints, clustering techniques were utilized to extract similar mobility behaviors. Sequential pattern mining techniques were used to identify mobility patterns from the movement histories of the users in the coverage area. The idea of clustering users with similar behavior reduces the complexity of the algorithm and consequently the computation time, and the combination with pattern mining techniques to discover frequent mobility patterns efficiently originates a methodical approach in mobility prediction. However, its accuracy depends on the size of the data and the complexity of the group within the region.

Movement Parameters

The purpose of this approach is to analyse the moving user's parameters and build a predictive algorithm through mathematical formulations or stochastic processes. The work in[16] shows an implementation of this process. Movement data and context information of diurnal user movements were exploited to predict cell transitions. This scheme has limitations, because if the cells were distributed randomly it would be difficult to obtain the cell transition, so a uniform distribution of cells is needed. Moreover, it depends on the prior knowledge of moving direction. Although this type of scheme is easy to implement it depends on the accuracy of the moving parameters and yet could not be sufficient to successfully perform prediction.

2.3.5 Artificial Neural Networks

Pioneer of artificial intelligence (AI), ANNs are a family of models designed to simulate the central nervous system of the human brain. These models have self-learning abilities that enables them to process better results even though the input is unknown to the network. Fig.4 illustrates a basic design of an ANN. Identical to the human brain, an ANN has thousands of artificial neurons named processing units interconnected by nodes. These represent the input and output units, the main objective of the network is to learn about the information that the input processing unit provides and construct an output to

submit to the output processing unit. The connections have unfixed numeric weights indicating their adaptability. An ANN initially goes through a training phase where it learns to recognize patterns in data and generates the "first"output. Afterward, it compares the output produced with the actual desired output, if there is confliction the network uses a training rule called backward propagation of error or in simple terms, backpropagation. This rule implies that the network works backwards, going from the output to the input. Throughout its retreat, the weight of its connection is adjusted until the difference between the actual and desired output constitutes the lowest possible error. A comparison between polynomial perceptron network (PPN) built based on Weierstrass approximation theorem and multilayer perceptron backpropagation neural network (MLP-BP) were provided in [17]. MLP-BP outperformed PNN however, the computation cost of PNN was better than MLP-BP as back propagation requires a large amount of computational time. To this day, ANNs are a good candidate for modelling a predictive system because of their remarkable properties of learning.

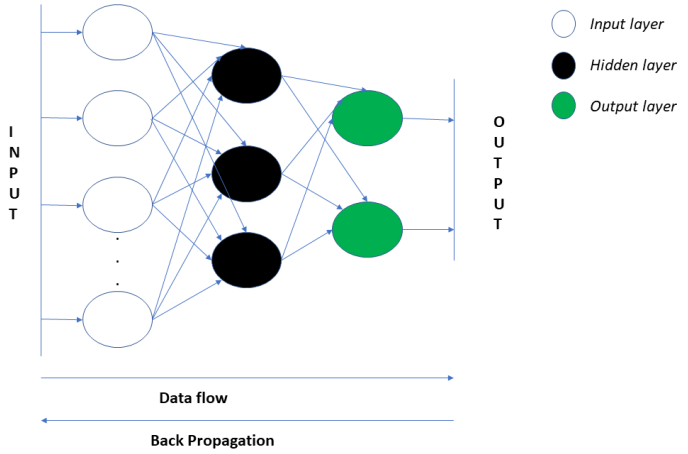

Figure 2.4: Basic design of an ANN.

2.4 Network Optimization Applications

With the demand for managing efficiently radio resources and the rapid progress of artificial intelligence, mobility prediction represents the most "sophisticated"tool in optimizing the network. It has a wide range of applications ranging from motion prediction approaches in vehicles, pedestrians and others, to cellular networks approaches essentially, next-cell prediction. In the next subsections we are going to discuss the aforementioned applications.

2.4.1 Vehicular Prediction

The breakthrough of vehicle-to-vehicle (V2V) communications will boost the design of intelligent vehicle applications like self-driving assistance, vehicle based sensing data collection, traffic safety, geo-advertising, in-vehicle internet access, and pothole detection [18][19]. Typically, the input of a vehicle prediction algorithm consists of the historical trajectory of the object vehicle during the last few seconds and the output is the predicted trajectory in the following seconds [20]. According to 2.5, vehicular prediction models are divided into two types: maneuver-based model and end-to-end model[20].

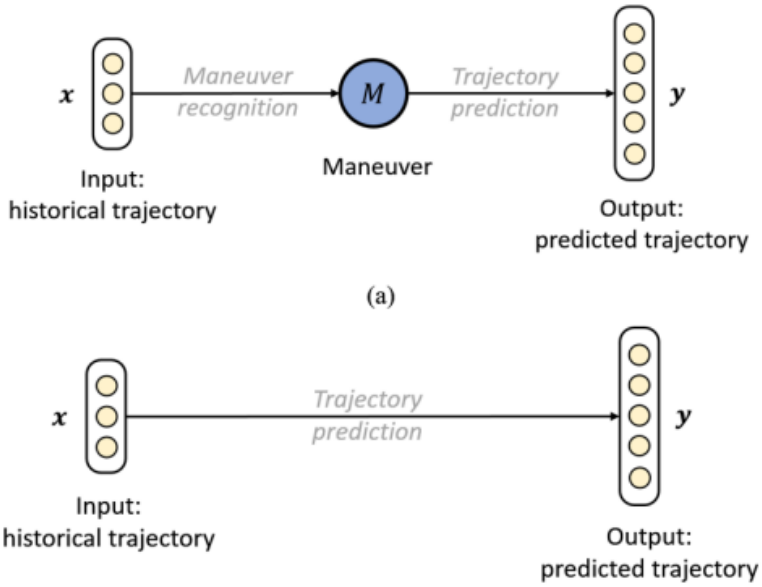

Figure 2.5: Vehicular prediction models: (a) Maneuver-based model (b) End-to-end model (taken from [20])

A maneuver-based model has an intermediate step named maneuver recognition step which outputs the motion maneuver of the vehicle. This information will be used and interpreted by the prediction step to generate the most accurate prediction trajectory of the vehicle. These models have an intuitive approach since maneuver categories are in accordance and from the knowledge of every human driver. However, labeling these maneuvers when training the model results in an computational expensive task and it is susceptible to incorrect maneuver labeling. End-to-end models bypass the maneuver recognition step and output the trajectory prediction directly. The work of [20] proposed a model named Spatio-Temporal-Long Short-Term Memory (ST-LSTM) to cope with the flaws of the end-to-end LSTM vanilla models. The authors used different LSTM models to handle the temporal relations and spatial interactions as different time series. These requirements implicate an addition of more layers in the network and when doing back-propagation, the gradient value can decrease over the layers ("vanishing") which can result in values too small for training to work effectively. This problem is known as the vanishing gradient problem. To overcome this adversity in the backpropagation, they

introduced shortcut connections between the input and output of every LSTM layer. Results showed that this model solved the two problems of long-term prediction in dense traffic. Under the scope of LSTM neural networks, [21] provided a personalized motion prediction algorithm based on a Gaussian Mixture Model (GMM) to classify driving styles and predict the trajectory of the leading vehicle. This Joint Time Series Model (JSTM) was compared to other prediction models such as Constant Kalman Filter (CKF), Multiple LSTM (MLSTM), LSTM and it outperformed every method. However, this model is limited as only tracks the leading vehicle, being an insufficient approach in a traffic awareness system. Another interesting vehicular application, is to incorporate the data provided by ITSs in order to build an enhanced large-scale traffic prediction algorithm. The work in [22] applied a temporal-window-based support vector regression (SVR) method based on k-means clustering algorithm to extract spatial performance patterns to find road segments with identical performance. This model showed success in predicting patterns in a large-scale traffic and it presented a viable solution in the development of route recommendation algorithms. Wei and Yozo [23] implemented a RNN-based vehicle mobility prediction algorithm named DeepVM to support intelligent vehicle applications. DeepVM outperformed other state-of-art algorithms such as Markov models-based. Its supremacy comes from its ability to process longer input data which reduces the uncertainty of the prediction.

2.4.2 Next-Cell Prediction

As mentioned in Section 2.3, the predictability of user mobility can be obtained at a solid value of 93% reflecting a cyclic pattern in human mobility behavior. However, the methods to achieve this value are dependant on context data and special movement patterns[24]. Instead of predicting user's accurate position, next-cell prediction algorithms offer an efficient strategy in dealing with handover connections issues. If wireless networks operators know which cell will the user enter they can optimize the network performance and enable a better QoS. The article of [24], classified mobility data for next-cell prediction into two types: Current Movement State (CMS), and Historical Movement Pattern (HMP).

Current Movement State (CMS)

CMS is based on real-time movement, meaning, the parameters evaluated are user's position, direction, acceleration, etc. Position and direction can be easily obtained through the UEs that can periodically communicate their location coordinates to BS. Based on their position coordinates, the BS can calculate the others motion parameters. The authors divide Current Movement State Based Approaches (CMSA) into three categories: angle-based, distance-based, and angle-distance combined ways. The survey both angle and distance should be considering when predicting, therefore angle-distance approach

results in the most effective way. Implementation of this method can be viewed in [25] where both angle and distance are used to predict TPs in UDNs.

Historical Movement Pattern (HMP)

HMP data is generated by observing users long-term movements and includes Cell-ID with CMS history, handover history, historical sectors, etc. By using CMS data, HMP updates its user's movement database, and with the appropriate prediction tools can effectively predict cell transitions. Historical Movement Pattern Based Approaches (HMPA) utilizes prediction schemes based on probabilistic models (e.g. Markov chain), discriminative models(e.g. ANN), and data mining. For example, the authors mentioned a Support Vector Machine (SVM) discriminative model type with an appropriate kernel function defined to find the support vector in the high-dimensional space by converting the non-linear input space into a high-dimensional space to form the optimal hyperplane. The authors of [26] implemented a multi-class SVM that considered a road topology scenario resembling an urban center, regular streets, and rail areas. The input was generated data from a mobility model of integrated path follower, gravity, and random walk models. Simulations showed that the prediction accuracy of the next-cell can exceed 90%.

Hybrid Approaches (HDA)

HDA assembles both CMS and HMP prediction techniques. While exploiting HMP to identify movement cyclic patterns, it can also perform a real-time estimation for instantaneous motion (CMS). Besides, it solves the short-term prediction problem of the CMSA and the high computational complexity issue of the HMPA. It is the most viable solution in dealing with short-term and long-term mobility prediction. In [27], for the long-term trajectory, i.e. cell sequence in weeks or months, the authors used a Markov update process to mine the regular movements. To deal with the randomness of short-term movement, they determined user's movement direction through the empirical moving average. At last they combined these two prediction results through Dempster Shafer theory.

2.5 Mobility Datasets

Datasets consist of a collection of data and are the main input of data analysis processes. They are represented by tabular data in a spreadsheet format where the rows are the records of the events and the columns are the characteristics of those events. So how to choose an appropriate dataset to study and evaluate mobility prediction? It depends on the approach and the type of mobility (trajectory oriented or cellular oriented). For example, motivated by the randomness and unpredictability of taxi routes, the work in [23] collected data of 65 taxis in Tokyo, Japan, in a period of four months. Both [21][24] used NGSIM I-80 and US-101 freeway data to build a model that could perform prediction

in a dense traffic flow situation. The authors of [22] extended the previous work by using data from a large subnetwork in Singapore that consisted of a diverse set of roads having different lane counts, speed limits, and capacities. Such variety of data enables pattern identification and construction of a more adaptable and general prediction algorithm. On the other hand, in [28] they used GPS data to identify atypical moving patterns in individuals. The paper concluded that the inclusion of atypical patterns enhances the prediction accuracy of their model in comparison with other state-of-the-art prediction methods. Another interesting type of collected data can be viewed in [29], the authors utilized smart card and points of interest data to perform prediction in bus travellers. Each record of the bus smart cards provided the origin and destination of the traveler. Dedicated to predicting mobility for IoT devices, in [30], to reflect the population effect they outlined a path during work hours and collected Wi-Fi data using a smartphone. Also, they used the Universal Software Radio Peripheral (USRP), a dipole antenna and a computer to collect the cellular data. The versatility of mobility prediction algorithms and the wide variety of datasets indicates that there is not a protocol nor a general rule in choosing an appropriate set of data to perform prediction. However, its choice will be directly correlated to the success of the model. Therefore, datum should be studied and evaluated thoroughly to guarantee the best accuracy prediction. Table2.1 provides a brief explanation on the different types of parameters that one should evaluate when choosing an appropriate dataset to perform mobility prediction.

Mobility dataset categorization	
Parameter	**Definition**
- Mobility type	It can be vehicular, bicycle or other movement related activity
- Trace type	It can be a real trace provided from real human activity or a synthetic trace from a built simulation
- Gps logs	Gps logs provide reliable information relatively to mobility information
- Scenario	Ranging from urban/ non-urban scenarios
- Number of traces	The number of traces reflects the quality and insight of the dataset. Predicting mobility in a dataset with few traces is harder than a dataset with plentiful traces

Table 2.1: Table that categorizes the different parameters when choosing a mobility dataset

3

Mobility Data Analytics

3.1 Mobility Scenario

The increase in motor vehicular traffic translates into an increase in pollution and chaos. Bicycling constitutes a viable option in the population's choice of mobility. Motivated by this and allied to the fact that developed countries are prioritizing and integrating cycling with mobile applications, the dataset chosen is based on crowdsourced bicycle trips. The data was taken from Münster in Germany, this which is classified as a major city due to having more than 100,000 inhabitants (300,000 approximately). Located in the North Rhine-Westphalia region, northwest Germany, possesses, in terms of topography, mainly flat features. The flatness allowed the establishment of a high-quality network of dedicated bicycle lanes [31]. Such lanes portray 39% of the city trips, homing more than 500,000 vehicles.

3.1.1 Data Collection

Participants were recruited to provide cycling data through the employment of printed posters, flyers, social media, and other advertising tactics. The data of 20 participants was gathered through the *Cyclist Geo-C* application during summer-autumn season of 2017. During this period, participants recorded each bicycle trip through the app and reported it with up to three tags upon terminus. To promote competition and motivation to the users to record the trips, the app had a virtual leaderboard of all the users in the area based on the amount of recorded trips done per day, in addition, each participant received €10 when the duties were finished [31].

3.1.2 Essential Dataset Features

The Dataset was stored in GeoJSON format [32] and each participant is described by the following metadata:

- *Device*: An unique ID that identifies the UE;
- *Altitude*: Geometric altitude in meters;

- *Latitude*: Geometric latitude in decimal degrees;
- *Longitude*: Geometric longitude in decimal degrees;
- *time_gps*: The recorded day time of the trip in the format: *'YYYY-MM-DDThh:mm:ss.sssZ'*.

There were other dataset fields such as speed, acceleration, etc. However, they are not relevant to this study. With the *device* ID, which is the "fingerprint"of the user, we can characterize and filter the trips of the users, and with the values of *latitude*, *longitude and altitude* we can map the position of the user. Finally, the *time_gps* field provides the information of the sampling time. All of these features combined grant detailed insights in the construction of the prediction model.

3.2 Analytics Methodology

The approach of this work consists in mapping the spatial positions (x, y) of the bicycles into a grid forming a matrix. A previous study of the city area was done in order to understand the disposition of the grid. Figure 3.1 illustrates the corners of the grid map. The points were pinned based on city limits and bike lane density.

Figure 3.1: Pinned points delimiting the city of Münster.

3.2.1 Problem Notation

The grid is divided into smaller two-dimensional rectangles named as cells, c_η. For each bike location, registered at a fixed sampling period, there is an associated cell. So,

by representing spatio-temporal displacements we can define trajectories, where each trajectory consists of the initial point of the journey until the endpoint of the journey. Finally, these trajectories are partitioned into smaller trajectories named sequences which denotes a fixed number of visited cells in the trip. Lets state the problem notation to define the proposed model.

- **Definition 1:** A cell c_η with $\eta \in \{1,... N\}$, represents a two-dimensional sub region of the grid. Each location of the bicycle is represented by a cell. N denotes the maximum number of cells forming the grid.

- **Definition 2:** A set of multiple cells is given by $T_j = \{c_\eta^1, c_\eta^2, c_\eta^3,...,c_\eta^{\Xi_j}\}$, and it is described as a trajectory. The bicycle travels in an ordered sequence of Ξ_j cells were $\Xi_j > 1$.

- **Definition 3:** $S_k = \{c_\eta^1, c_\eta^2, c_\eta^3,...,c_\eta^\Lambda\}$ represents an ordered sequence shaped by a set of Λ visited cells, where $\Lambda \leq \Xi$. It is important to note that Λ remains constant through the space state Ω, where Ω defines all sequences from a given trajectory.

- **Definition 4:** A set of sequences is expressed by $\Phi = \{S_1, S_2, S_3..., S_\Omega\}$, where Φ is a set constituted by all Ω sequences. Since bicycles trajectories have a high degree of randomness there is a necessity of express unique sequences. Therefore, the unique sequences in Φ are represented by the symbol Ψ.

- **Definition 5:** The estimation problem uses empirical observation knowledge of Λ - β cells in a sequence S_k to estimate/predict the next cells of the sequence, with $\beta \geq 1$.

Table 3.1 summarizes the symbols used in this work.

Problem Notation Classifier	
Symbol	**Description**
N	Number of cells that constitute the grid
C_η	Cell η, $\eta \in \{1,...,N\}$
Ξ_j	Trajectory defined by j cells
S_k	Sequence defined by k cells, where k $\in \{1,...,\Omega\}$
Ω	Total of trajectory sequences
Λ	Total of sequence cells
β	Number of cells to predict based on the observation Λ - β
Φ	Set of all sequences
ϕ	Unique sequences in the set Ψ

Table 3.1: Table with the symbols and descriptions used in the prediction problem.

3.2.2 Data Preprocessement and Characterization

With the support of the polyline entry of the dataset relative to the GPS coordinates, we filtered the raw GeoJSON data to inscribe the values of latitude and longitude of the city. Next, we identified each trip with the device entry of dataset enabling ordered and unique journeys. The value of the cells formed a file text were each line of the file represents a trip. If the difference of two consecutive times is 1000 seconds (approximately 16 minutes) a new line is added to the file resembling a new trip. Subsequently, a time sampling value of 30 seconds was fixed and the GPS coordinates were transformed into Cartesian and mapped into a cell, cη, i.e., $1 \leq \eta \leq 16$, as illustrated in Figure 3.2. After the mapping of every spatial positions into a cell, we built trajectories where, $T_j = \{c_\eta^1, c_\eta^2, c_\eta^3,...,c_{\eta\, j}^{\Xi}\}$, has a variable number of locations given by Ξ_j cells. The next process is to cluster the cells of a trajectory to form a sequence, logically, each trajectory can have more/none sequences. The parameter that determines the previously mentioned is Λ, where a sequence $S_k = \{c_\eta^1, c_\eta^2, c_\eta^3,...,c_\eta^\Lambda\}$ is constituted by Λ cells. The values of Λ used in this work are constant during the prediction process. The grids used in the project were with N = 16, N = 64, N = 256. Their area and cell area are depicted in Table 3.2.

Figure 3.2: Cell disposition for a grid with N = 16 (4x4)

N	Grid Area(m^2)	Cell Area(m^2)
16	4818171	301136
64	1204543	18821
256	301136	1176

Table 3.2: Grid area and cell area for different values of N.

Logically, as N increases the grid and cell area decrease. This matter is critical in the preprocessement of the data, why? Because if we analyze the cell positions of a user in a smaller grid such as 4x4 (N = 16) they will differ not only in number but also in occurrence, concerning bigger grids such as 8x8 (N = 64). If we have a bigger grid, the cell area is smaller meaning that the probability of a user being in one or more cells that make up a cell in smaller grids is higher. As we escalate the grid size, the existence of different trajectories patterns is increased resulting in a dynamic feature to take into account. We can study and interpret the behavior of the bicycles as their cell sampling position is going to affect immensely the relation grid/trips and the generation of unique sequences. In the next section, a more in-depth analysis of the aforementioned behavior is going to be discussed, and the neglect of bigger grids (i.e. 1024, 4096, etc...) is going to be addressed.

3.3 Dataset Analytics

With the incorporation of the grids, the next objective is to form valid sequences from the raw data. Therefore, Figure 3.3 contains the flowchart that generates the sequences from the trajectories Ω. The file possessing the trips of each user and the value of Λ

indicating the length of the sequence is loaded as input. Reasonably, the value of Λ should be greater than 1, if so, then we initialize k with the unitary value, where k serves as an index of the sequence initially empty S_k. Next, each line of the trajectory file, represented as T_j, is looped using a sliding window ($n\_window = 1$) until the end-of-file (EOF) is reached. The use of a sliding window plays a crucial role in developing sequences from trajectories since it checks each cell in the trajectory line of the file. In the sliding window loop, the variable i is initialized and constitutes the index of each cell in the trajectory file. Another cycle is introduced forming a nested loop, where each cell is iterated continuously up to the nonexistence of the cell in the trajectories array. Every cell in the previously mentioned array is appended to a list that will construct the sequence S_k. Finally, if the index of the cell surpasses the predefined value of Λ, k is incremented to form a new sequence, the list S_k is emptied, the value of i is set to search for the length of the sequence and the sequence is appended to the set of sequences.

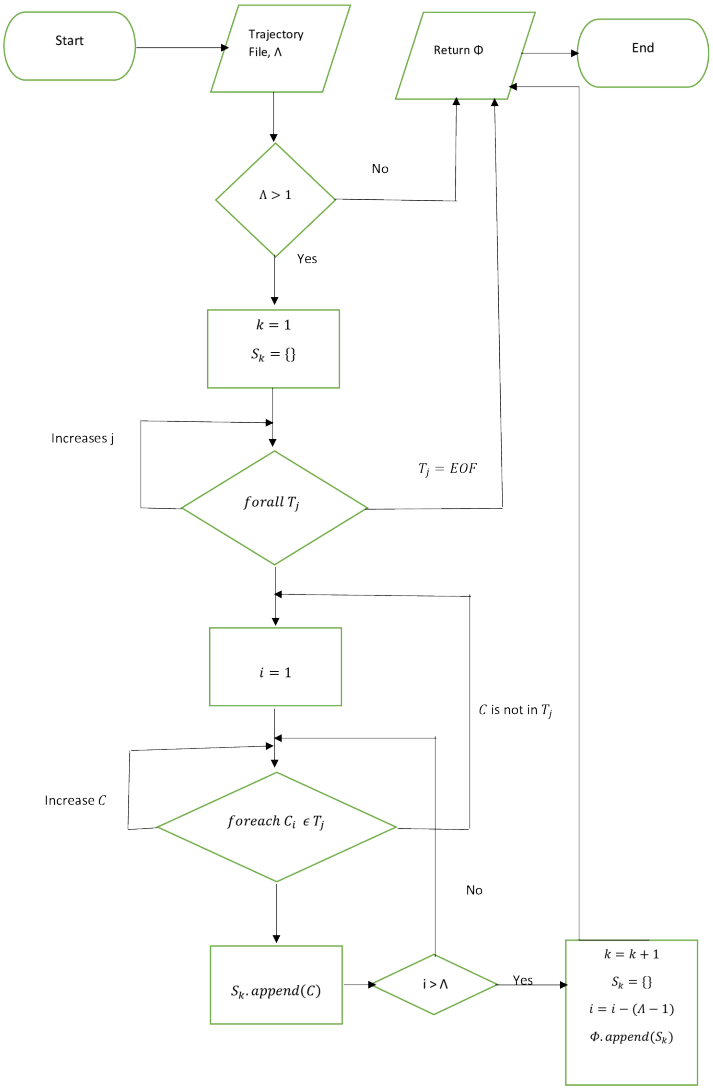

Figure 3.3: Flowchart that describes the procedures to generate sequences.

3.3.1 Sequences Cumulative Distribution Function

This subsection examines the weight of the observation space which characterizes the sequences Φ and the unique sequences ϕ. Table 3.3 outlines the preceding sequences for diverse values of Λ, 4, 8, 12, and N, 16, 64, 256. We can decipher a inverse probability relation between the number of trajectory sequences Ω and the number of unique sequences ϕ, caused by grid escalation. As the grid increases, Ω decreases and ϕ increases, this phenomenon is explained by the mobility of the bicycles which is the same in all grids, and the consequent reduction in cell area, whereby creates different patterns in sampling the cell positions.

N	Λ	Φ	Ψ
16	4	5126	187
	8	4424	524
	12	3797	812
64	4	5126	468
	8	4424	1233
	12	3797	1795
256	4	5126	513
	8	4424	1307
	12	3797	1863

Table 3.3: Table containing the value of the observation space Φ, unique sequences ϕ, for different values of grid and sequence length Λ .

3.3.1.1 Sequences CDF Analysis

To compute and specify the incidence of the unique sequences ϕ that contemplate each S_k sequence we used the Cumulative Distribution Function (CDF).

- **Definition**: CDF serves as a probability tool to represent the probability distribution of random variables defined as:

$$F_X(x) = P(X \leq x) \tag{3.1}$$

The equation 3.1 denotes the probability of a random variable X being lesser than or equal to x. To contextualize and integrate in our work we computed CDFs for various Λ and N values. In Figure 3.4 we visualize that the y-axis designates the cumulative probability of the first 100 sequences in S_k. The isolated dots represent the unique sequences with more probability and incidence, and the stabilization of the function is explained by the cumulative sum of the less probable unique sequences occurrence.

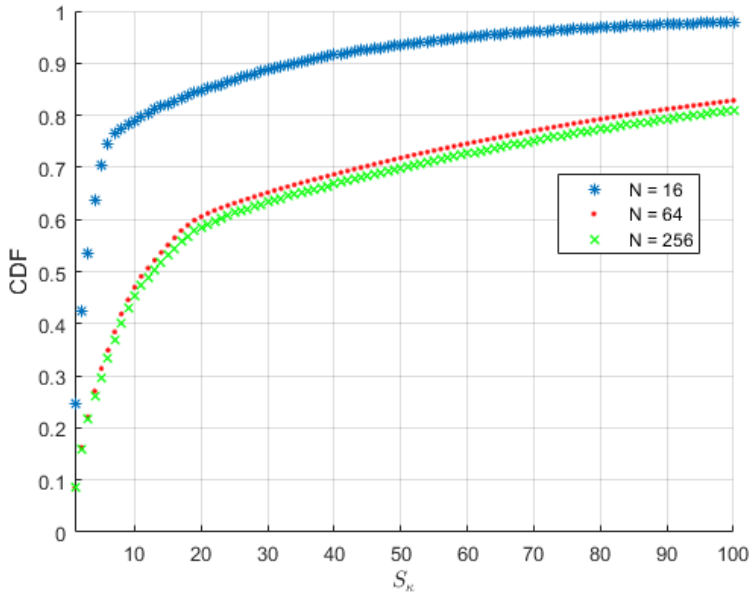

Figure 3.4: CDF with $\Lambda = 4$ for various N.

The above figure illustrates a CDF with sequence length $\Lambda = 4$. We can elucidate that a low value of lambda such as 4 conducts to a high percentage probability of unique isolated sequences and less unique sequences in ω. Nevertheless, with a increase in the grid to N = 64 or N = 256 the stabilization of the slope occurs in lower probability values resulting in an increase in unique sequences. Even though, we can obtain interesting values to study prediction they are not challenging enough, hence Figure 3.5 and Figure 3.6 demonstrate better CDF values to study the influence of unique sequences.

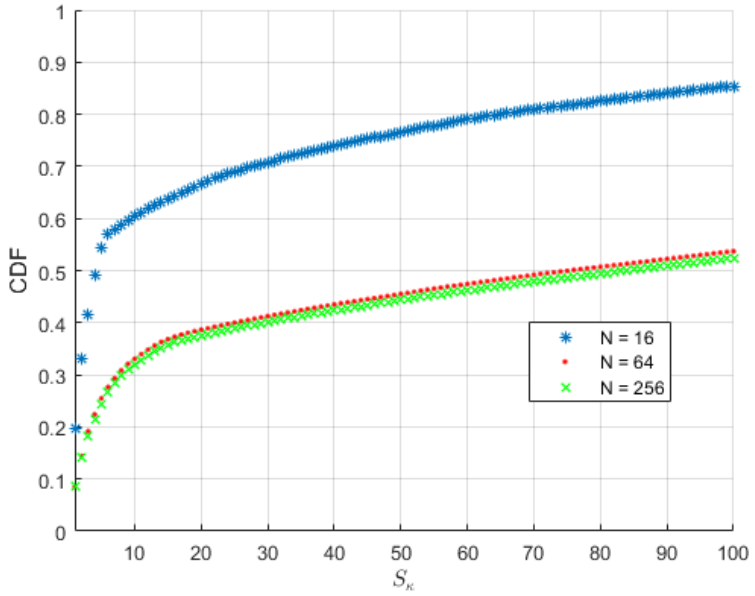

Figure 3.5: CDF with $\Lambda = 8$ for various N.

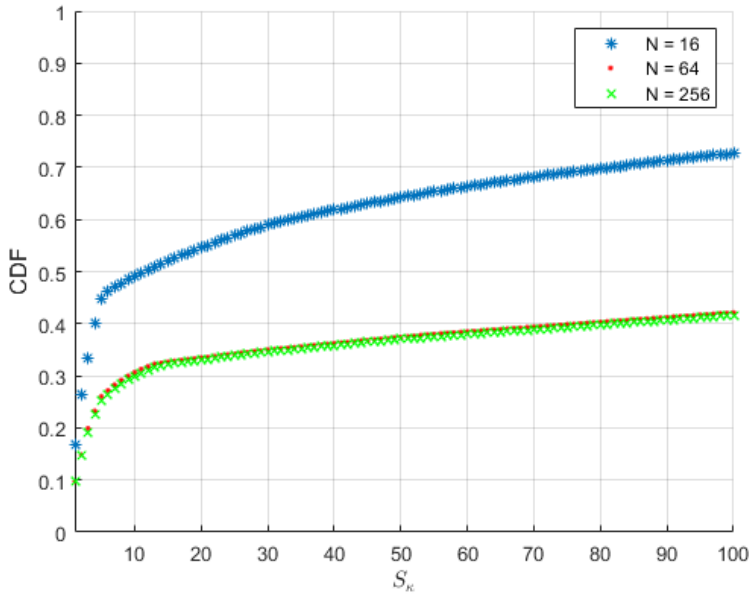

Figure 3.6: CDF with $\Lambda = 12$ for various N.

As we can depict, an increase in Λ means an increase in the length of the sequence where the appearance of unique sequences is more probable. Additionally, the increase in the grid and the consequent reduction in cell area are a plausible reason for the existence of more unique sequences. All of these values are in accordance to Table 3.3 were we can analyse the effect of the growth in Λ and N on the sequences. In terms of the prediction problem, more unique sequences implicates more randomness which adds a more compelling twist in the architecture of the algorithm, so why not use bigger grids and sequence lengths such as N = 1024, N = 4098, $\Lambda = 20$, $\Lambda = 24$ etc...? Mathematically this should produce more unique sequences, although with this datum, the increase of these parameters did not bring relevant CDF values that justified its conceptualization.

4

Deep Learning Architectures

In the next chapter, the approach leading to the construction of Deep Learning Architectures will be discussed. Firstly, the different scenarios will be shown and the respective implication in handling the data with the one-hot encoding solution will be explained. Next, an in-depth explanation of the models LSTM and ANN will be provided, and finally, a performance evaluation and comparison of both neural networks will close the chapter.

4.1 Scenarios and Data Handling

With the sequences constructed and the incidence of the unique sequences studied by the CDF's evaluation, the objective that follows and ultimately the main objective of this work is to predict the next grid cell of the cyclist. To perform that we have to understand the scenarios of the problem which are characterized by the following parameters:

- **Three different grid sizes:** N = 16, N = 64 and N = 256;
- **Three different Λ values:** $\Lambda = 4$, $\Lambda = 8$ and $\Lambda = 12$;
- **Multiple outputs/cells to determine:** The outputs to predict will be $\beta = 1$, meaning only the last cell of the sequence will be estimated, $\beta = 3$ outputs, meaning the last three cells of the sequence will be estimated, and $\beta = 4$ outputs, defining that with the observation knowledge of a predetermined length of a sequence, the last four cells will be estimated (consult definition 5 of the problem notation in Section 3.2).

.

Let us evaluate each scenario for the three sequence lengths (Λ):

Scenario $\Lambda = 4$

$\Lambda = 4$ means that the length of the sequence S_k will have four cells. The input size of the data is 5126 sequences for all three grids and the desired output is the last cell per sequence $\beta = 1$, knowing Λ - $\beta = 3$. In this case is not interesting to test for $\beta = 3$ because Λ - $\beta = 1$, one cell of knowledge is not sufficient to obtain good prediction results. Logically, the case $\beta = 4$ is also not taken into consideration.

Scenario Λ = 8

$\Lambda = 8$ indicates a length of eight cells in S_k. The input size of the data is 4424 sequences for all the three grids and the desired output is the last cell per sequence $\beta = 1$, knowing Λ - $\beta = 7$, and the last three cells per sequence $\beta = 3$, knowing Λ - $\beta = 5$. Reasonably, the last three cells can be determined with success due to the longer length of the sequence.

Scenario Λ = 12

$\Lambda = 12$ represents a length of twelve cells in S_k. The input size of the data is 3797 sequences for all three grids and the desired output is the last cell per sequence $\beta = 1$ and the last four cells per sequence $\beta = 4$, knowing Λ - $\beta = 8$. As a result of a longer sequence length, this scenario has the longest output cell prediction.

Table 4.1 summarizes the aforementioned parameters scenarios:

N	Λ	Φ	Λ - β
16	4	5126	1
	8	4424	3
	12	3797	4
64	4	5126	1
	8	4424	3
	12	3797	4
256	4	5126	1
	8	4424	3
	12	3797	4

Table 4.1: Table indicating the main parameters of the work and the respective inputs and outputs for each scenario.

4.1.1 One-Hot Encoding

One-Hot Encoding is a machine learning binary technique used to convert data in order to enhance the performance of an algorithm. In simple terms, we assign a vector of 0's and in a specific index a 1 to categorize data. Different values of the data will have different binary vectors combinations with a unique unitary value in a given position of the vector. Transposing into this work, the one-hot technique is applied to distinguish and categorize the cells of the grid. For example, Figure 4.1 illustrates an array with 17 columns representing the case of 16 cells in a grid and 3 lines representing the knowledge observation we get for $\Lambda = 4$, we can visualize that the first cell is 15 because of the unitary value present in the index 15. Additionally, cells 2 and 3 share the same cell represented by a completely different combination of 0's and 1 indicating cell 11.

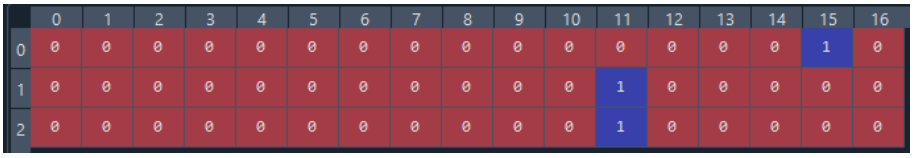

Figure 4.1: Illustration of the one-hot encoding technique in 3 cells of a sequence.

The orthogonality that this technique provides to the data is an added value in the development of the model since it permits better expressibility and guarantees a rescale central property in the manipulation of big data.

4.1.2 Model Assumptions

The models used in this work are 2 deep learning models. These models are trained based of the dataset to further execute the prediction process. The models utilized are an Artificial Neural Network (ANN) and a Long-short Memory Network (LSTM). The variables that are previously prepared to enable the network training are the length of the sequence Λ, the number of cells N that the network takes into consideration, and the number of cells β that the network should predict. These variables and their values are common in both network cases, however, the main core parameters of the networks are common but their values differ. Their definition is the following:

- **Epoch:** As the name suggests, is an epoch of training the neural network with the information of the dataset. The number of epochs that a neural network contains indicates the number of forward and backward passes that the network executes. An epoch is organized in batches and the number of batches is passed through each epoch.
- **Batch Size:** The data is organized in batches to perform training. The number of samples placed in the batches constitutes the batch size used in each epoch to perform the training.
- **Input Layer:** The Input Layer is composed of units or neurons and is the gateway into the network. The data enters through this layer and the processing starts stimulating the data to be passed to deeper layers.
- **Hidden Units:** The Hidden Units comprise the subsequent layers of the Input Layer and they drive and process the information to the Output Layer.
- **Output Layer:** Is the last layer of action formed by a fixed number of units in which the desired output is derived through this layer.

In both cases, the number of cells is the same, from this number the sequences are built obeying the Λ value and these sequences are transformed into one-hot encoded vectors that serve as input to the networks. The output is a one-hot encoding matrix that

computes the predicted β cells for each one-hot sequence. In the next section, we discuss and explain the main differences and the *modus operandi* of both scenarios.

4.2 Artificial Neural Network Learning Model

The first model that we present to solve the prediction of the next cell/cells problem is the "mother"of artificial intelligence and it is an ANN base type architecture. We built the ANN with 4 layers: The first corresponds to the input layer and it has (Λ - β) * N units and its input shape is carved to the same dimension. Thus, the information that the network knows in the input layer is in accordance with the empirical knowledge observation in relation to the grid size. The activation function used to activate the neurons is the Rectified Linear Activation Function or in short ReLU. The definition of this function is given by:

$$f(x) = max(0, x). \tag{4.1}$$

In addition to being a computationally simpler function, the linear behavior that imposes allied to the fact that the encoded input information is constituted by 0's and 1's we can easily see why it is the most compelling activation function for this particular case. Plus, to top it off it solves the vanishing gradient problem that other activation functions suffer in the ANN realm. Digging into the network, we have 2 hidden layers that are comprised by the same units of the input layer with the twist that they decrease, in the first hidden layer, by a factor of 80%, and in the second by 60%. These values were arbitrary (only following the premise that they should be smaller than the input units) and they produce better training results. The activation function on these layers is also ReLU. Finally, the output layer is composed of the output units of, logically, β * N, as we want to estimate β cells in the grid of N size. The activation function used in the output layer is the Softmax function, why Softmax? Because it forces the output to represent the probability of the data being a defined one-hot vector which in turn represents a specific cell in the grid. Hence, figure 4.2 represents a summarized ANN architecture that receives an input one-hot matrix and outputs a one-hot encoded matrix where the lines are β and the columns are N.

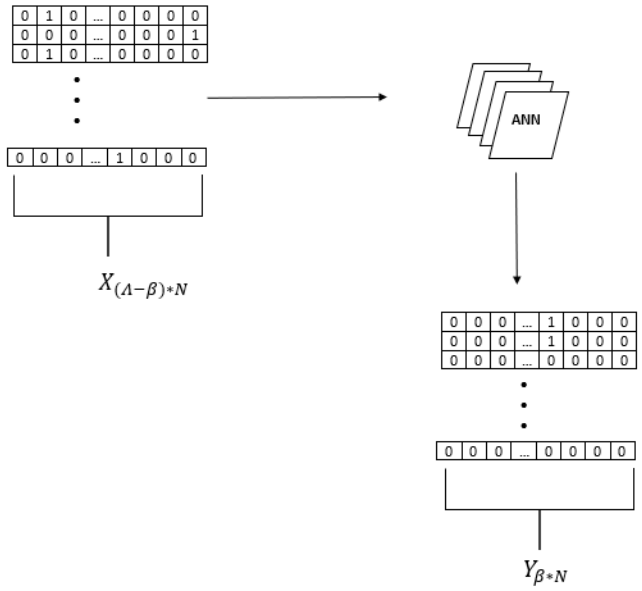

Figure 4.2: ANN input and output structure.

In regard to compilation training parameters we used the*Adam* optimizer and *Categorical Crossentropy* as loss function since harmonizes well with the one-hot encoded data. Table 4.2 summarizes the architecture components.

Layers	**Units**	**Activation**
Input Layer	(Λ - β) * N	ReLU
Hidden Layer 1	(Λ - β) * N	ReLU
Hidden Layer 2	(Λ - β) * N	ReLU
Output Layer	β * N	Softmax

Table 4.2: ANN layers configuration.

4.3 Long-Short Memory Network Learning Model

A Long-short Memory Network is a special type of Recurrent Neural Network (RNN) capable of learning entire sequences of data. The term memory is very important in this network and therefore let us understand how the memory particle functions:

- **Short-term memory:** Acts in the information acquisition aspect and is retained for a few seconds. The information then has two possible outcomes: Either is cached for longer periods or completely discarded.

- **Long-term memory:** Holds the information enabling its recovery or even to be used *a posteriori*.

How can LSTM control and select the flow of information that receives? Well, due to the fact that has a special gate (besides the normal input and output gates input) that

revolutionized the deep learning world and it is the main reason that these networks can manage the information in a critique manner, it is named **Forget Gate**. The forget gate decides which information is discarded/forgotten by the network. The interdependency in the layers and neurons enables the trajectory pattern recognition task to be flawless. There are many advantages in relation to the ANN that can already be foreseen nonetheless let us first explore the main architecture used in this work to estimate future cell positions. The LSTM we built has a simple structure with two layers functioning as input layer and output layer inspired by the work of [33]. Is in the hidden units where the effects of the input gate, forget gate, and output gate manifest. The number of hidden units we used is 16 shaped to support bi-dimensional data with β - Λ and N size. First, we thought that the hidden units should sample the size of N but promptly after some tests, we realized that 16 was the best value in the aspects of computation complexity and prediction results. The activation function of these units in the input layer is the hyperbolic tangent activation function (tanh) which takes any real value in the scope [-1, 1] fitting in the one-hot encoded data. The output layer has the same number of neurons as the ANN which is the number of prediction cells we want to estimate: β * N. The activation function of these neurons is the Sigmoid and it works in a similar fashion as the tanh, the difference being is that is a logistical function working with outputs in the intervals of [0, 1] values, which is coherent with the one-hot output. As depicted in Figure 4.3, the network receives one-hot vectors labeled $\mathbf{X}_i$ until the observation knowledge of β - Λ, is reached. These vectors are fed into the hidden units in which the flow of information of the trajectory patterns is managed, controlled and ultimately transported to the output dense layer. In the output layer we want to output β cells analogous to the ANN scenario. The output is given by $\mathbf{Y}$ which is a one-hot encoded matrix with the prediction information of the β cells per grid size N.

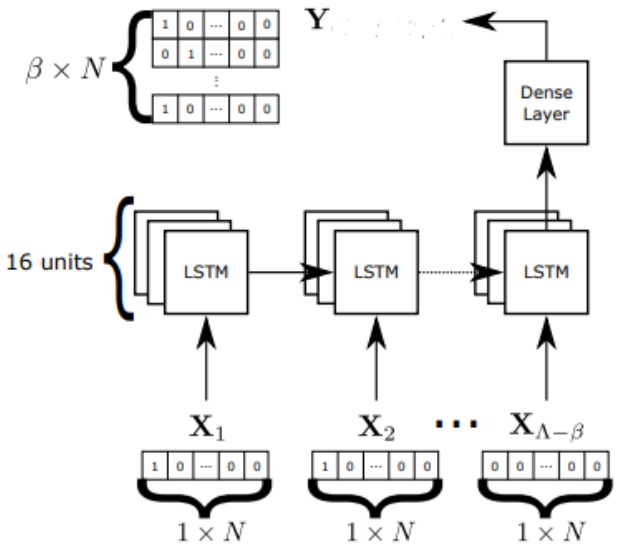

Figure 4.3: LSTM architecture (adapted from [34]).

In this architecture, the *Adam* optimizer and the *Categorical Crossentropy* function were also used so that we could have a valid comparison term of both models. The specifications of the network are described in Table 4.3.

Layers	**Dimension**	**Activation**
Input Layer	16	tanh
Output Layer	β * N	sigmoid

Table 4.3: LSTM layers configuration.

4.4 Performance Evaluation

In the next section, we are going to analyze in detail all the results produced in both deep learning architectures. First, the loss theme in the training phase will be discussed to then introduce and debate the prediction results where it will be divided into two types of prediction: Short-term prediction, refers to the $\beta = 1$ case, and long-term prediction relative to $\beta = 3$ and $\beta = 4$ results. The training and prediction of the models were executed using Python programming language with the support of Keras, TensorFlow, and NumPy packages. The MATLAB programming languange was used to gather all the results provided by the models. All the results and implementations were computed in an Intel 4-Core i7-8550U CPU @ 3.79 GHz computer with 16 GB of memory plus 8GB of DRAM. In [35] we made available a prototype of the project uploaded in a Github repository containing all the developed files/scripts of the work implemented.

4.4.1 Training evaluation

The training phase was executed using a batch size of 64 to comprehend the dataset of 4500/5000 average data. With this small batch size value, we had to exceed the epochs number and 1000 epochs were the defined value for training the network. To optimize and not run the risk of over-fit the model we defined 10 levels of patience. This means that the network will wait for 10 epochs before stopping the training if no progress is done. We have already mentioned that the loss function used in both scenarios was a crossentropy based categorical function. This loss function represents the difference between the actual processed data and the data processed by the training model. This difference generate the so-called loss function and the objective is to have a convergent loss function. For the example scenario of N = 16, $\Lambda = 4$ and $\beta = 1$, we can observe in figure 4.4 and figure 4.5 the required convergence aspect. If these functions diverge the trained data would not be in accordance with the real observable dataset which would produce bad loss values and ultimately poor prediction results. The convergence guarantees a decrease in loss values and correct training of the dataset.

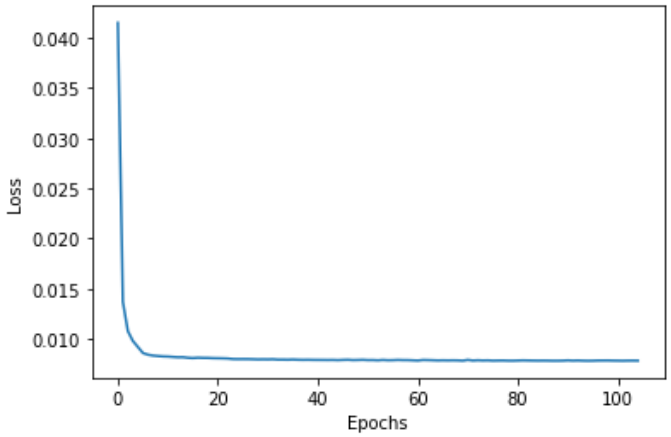

Figure 4.4: Categorical crossentropy loss function for a grid with 16 cells, $\Lambda = 4$ and $\beta = 1$ in ANN training.

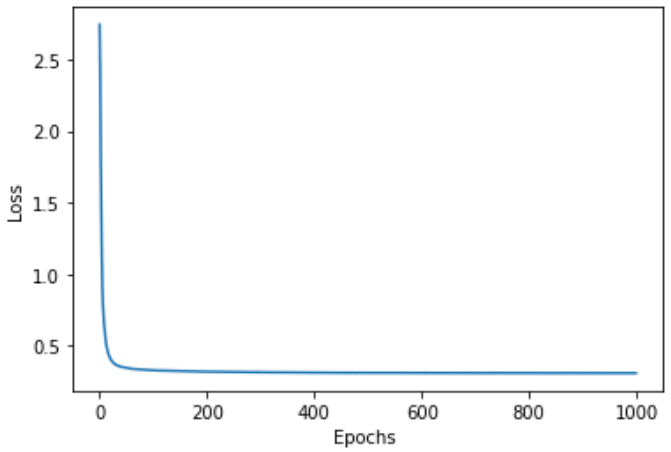

Figure 4.5: Categorical crossentropy loss function for a grid with 16 cells, $\Lambda = 4$ and $\beta = 1$ in LSTM training.

The graphics in Figure 4.4 and Figure 4.5 express the training success in short-term prediction, Figure 4.6 and Figure 4.7 illustrate the same convergence property for long-term prediction.

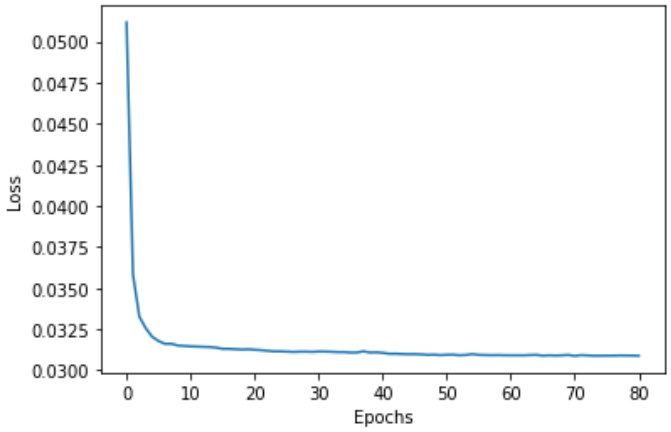

Figure 4.6: Categorical crossentropy loss function for a grid with 16 cells, $\Lambda = 8$ and $\beta = 3$ in ANN training.

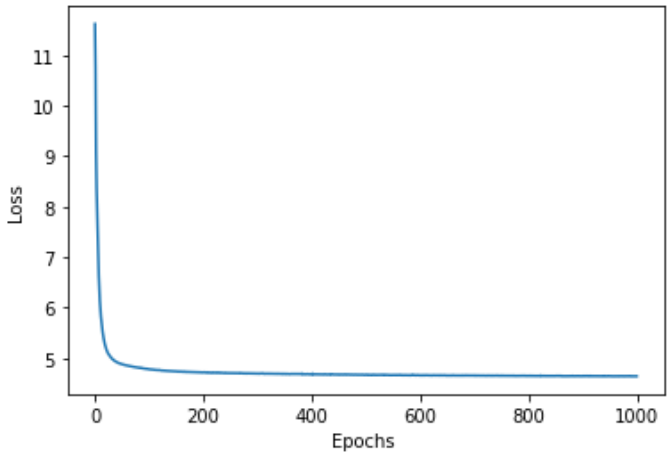

Figure 4.7: Categorical crossentropy loss function for a grid with 16 cells, $\Lambda = 8$ and $\beta = 3$ in LSTM training.

We can state that this property is present in both training models and in both short and long-term prediction cases. We can also assure that in the other scenario cases presented in Subsection 4.1 the same convergence slope occurs. With the training of the models well implemented and validated, we can enter the realm of the prediction results.

4.4.2 Prediction Performance

As we mentioned the objective of this work is to correctly implement a model that can estimate cell positions whether for a short instance ($\beta = 1$) or for long periods ($\beta = 3$, $\beta = 4$). All the results described in this section for these cell intervals are applied to all the grids as well as all sequence lengths described in section 4.1. The predictive models, ANN and LSTM, that encapsulate all the conditions of the problem are evaluated through

a study metric of Prediction Performance, as follows:

$$PP = \frac{1}{\Pi} \sum_{j=1}^{\Pi} G(c_{pred\,j}, c_{ori\,j}). \tag{4.2}$$

In equation 4.2 Π defines the total of sequences, where $G(c_{pred\,j}, c_{ori\,j})$ denotes a binary function that returns 1 when $c_{pred\,j}$, the cell predicted, matches $c_{ori\,j}$, the original cell in the sequence, and returns 0 if their value is different. With the sum of all correct/incorrect predictions divided by the number of sequences evaluated, we can compute the percentage performance of the models. To facilitate the display of results we are going to first assess the short-term prediction results and then debate the long-term performance of both models.

4.4.3 Short-Term Prediction

The short-term prediction is defined as the prediction of the last cell of a sequence ($\beta = 1$). All sequence lengths are considered for all the grids' sizes.

ANN and LSTM performance for $\Lambda = 4$, $\beta = 1$

Figure 4.8 and Figure 4.9 illustrate the prediction performance for both ANN and LSTM values for the sequence length value of $\Lambda = 4$ in all 3 grids, N = 16, N = 64 and N = 256. As we can observe by the bar graph the results of both models are practically equal. For the smaller grid the value is approximately 93% and for the remaining is about 87% demonstrating successful results in the first study case. This however can be explained by the high probability of occurring certain sequences (illustrated by the CDF's plots in Subsection 3.3). As this scenario have the biggest number of sequences ϕ and the lesser number of unique sequences ψ it is normal that both models will correctly evaluate the next cell position translating into high values of performance.

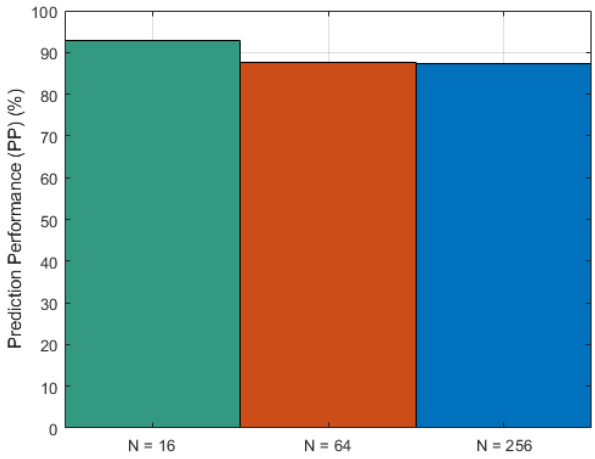

Figure 4.8: ANN prediction performance for $\Lambda = 4$, $\beta = 1$ and N = 16, 64 and 256.

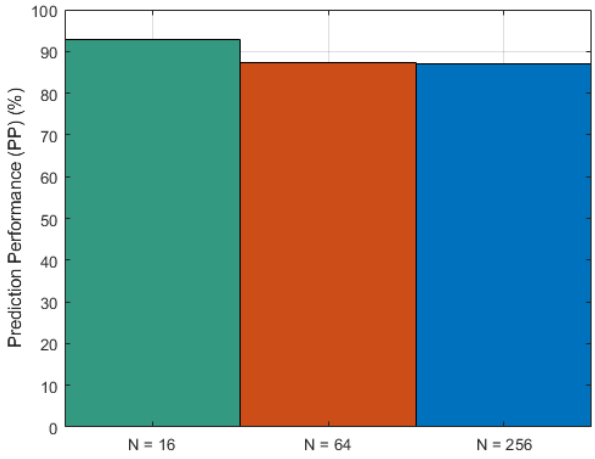

Figure 4.9: LSTM prediction performance for $\Lambda = 4$, $\beta = 1$ and N = 16, 64 and 256.

ANN and LSTM performance for $\Lambda = 8$, $\beta = 1$

With the increase in Λ the prediction performance will also increase as is depicted in figure 4.10 and figure 4.11. Why does this phenomenon happen? Because the models have more knowledge from the observation of more cells. Consequently, their inference and estimation capabilities will increase reflecting in values of 90% minimum.

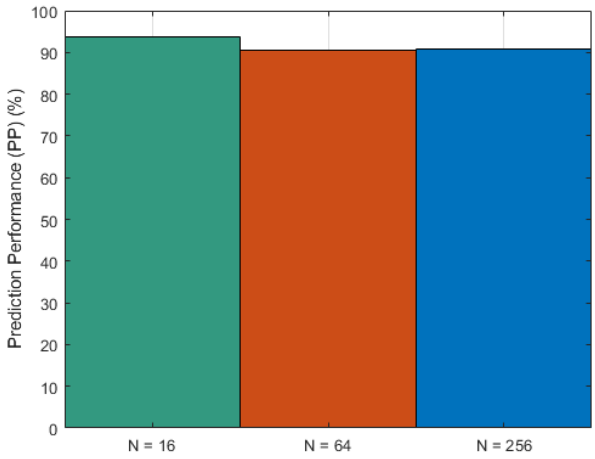

Figure 4.10: ANN prediction performance for $\Lambda = 8$, $\beta = 1$ and N = 16, 64 and 256.

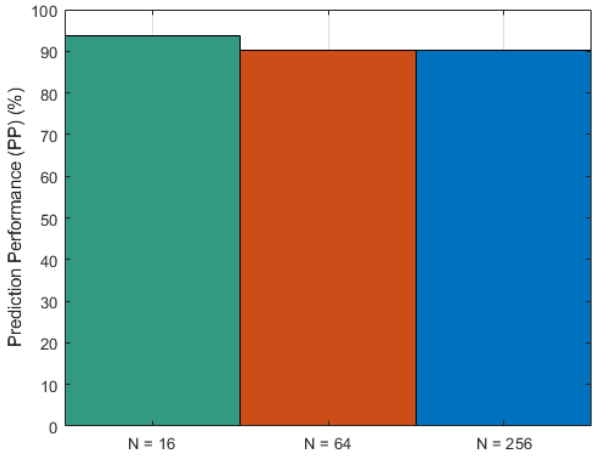

Figure 4.11: LSTM prediction performance for $\Lambda = 8$, $\beta = 1$ and N = 16, 64 and 256.

ANN and LSTM performance for $\Lambda = 12$, $\beta = 1$

Finally, the last case of short-term prediction represents the case where both models gather more observation knowledge. Albeit, this scenario produces more unique sequences resulting in a higher degree of unpredictability the vast knowledge that the networks possess allows high-quality estimation results represented in figure 4.12 and figure 4.13. These results recognize the power of both deep learning models where rates of 93/94% were achieved surpassing the previous scenarios and closing the sequence of short-term prediction impressively.

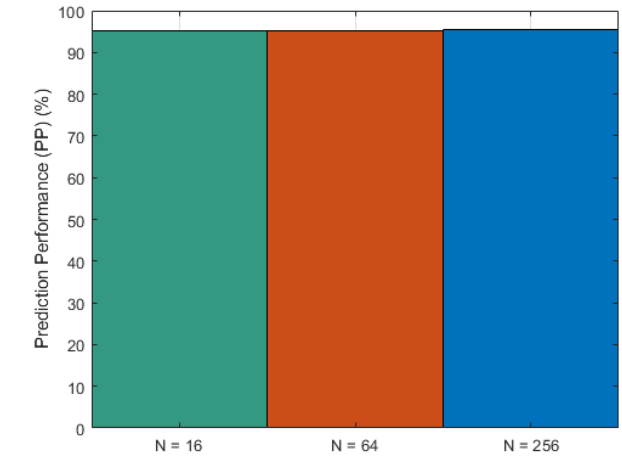

Figure 4.12: ANN prediction performance for $\Lambda = 12$, $\beta = 1$ and N = 16, 64 and 256.

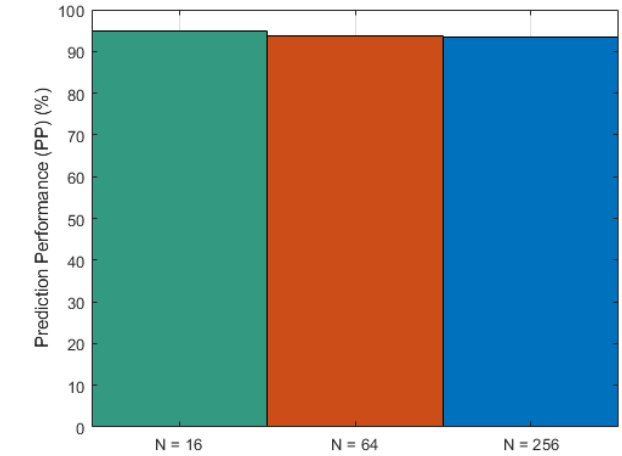

Figure 4.13: LSTM prediction performance for $\Lambda = 12$, $\beta = 1$ and N = 16, 64 and 256.

4.4.4 Long-Term Prediction

In the short-term prediction performance subsection, there were no relevant differences between the results of the ANN and LSTM models. However, in the long-term domain, the results are quite differ. The proposed cases are both for ANN and LSTM for $\Lambda = 8$ and $\beta = 3$ for all grids, and $\Lambda = 12$ and $\beta = 4$ also for all grid sizes.

ANN and LSTM performance for $\Lambda = 8$, $\beta = 3$

In this case, the architectures have the knowledge of 5 cells and need to correctly determine cells c^6, c^7, c^8. As we can see in Figure 4.14 and 4.15 in the next cell prediction,

c^6, the results are identical but as we go deeper in prediction, c^7 and c^8 the LSTM architecture outperforms the ANN architecture. There is not a particular reason rather reasons and aspects of the network such as the forget gate and hidden units, conjugated with the type of categorical information that enables better high probability prediction results. We can identify that in long-term prediction the escalation of the grid translates into a decrease in results, the reason being is the increase in the unique sequences ψ which generates more unpredictability and allied to a bigger grid develops a harder estimation task for the models.

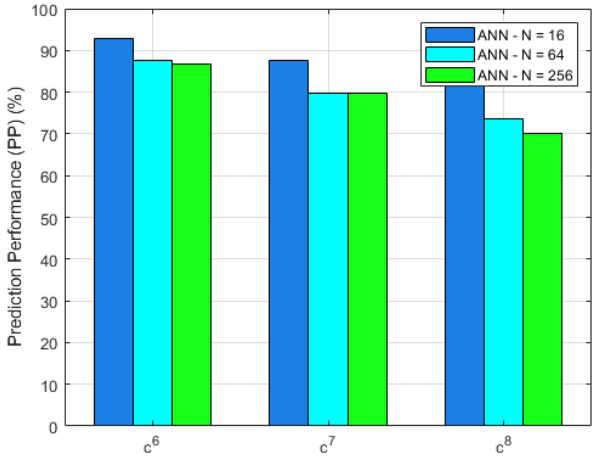

Figure 4.14: ANN prediction performance for $\Lambda = 8$, $\beta = 3$ and N = 16, 64 and 256.

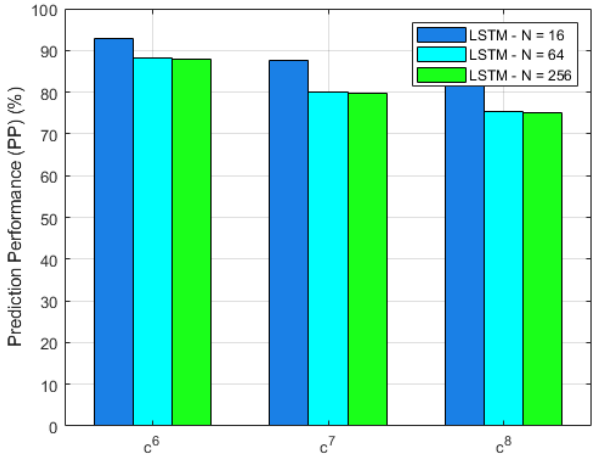

Figure 4.15: LSTM prediction performance for $\Lambda = 8$, $\beta = 3$ and N = 16, 64 and 256.

ANN and LSTM performance for $\Lambda = 12$, $\beta = 4$

The final long-term scenario enables the models to have knowledge from the observation state of 8 cells and require to correctly identify 4 cells c^9, c^{10}, c^{11}, c^{12}. As in the previous case, Figure 4.16 and Figure 4.17 portray a similar result in the next cell prediction c^9. The disparity of results in deeper estimations is similar emphasizing more in the bigger grid N = 256 where ANN estimated the last fourth cell with a probability of 53.2% and the LSTM computed a rate of 80.1%. Therefore, the task of longer predictions in bigger grids is extremely difficult and the capabilities of LSTM models excel from ANN models in these prediction scenarios.

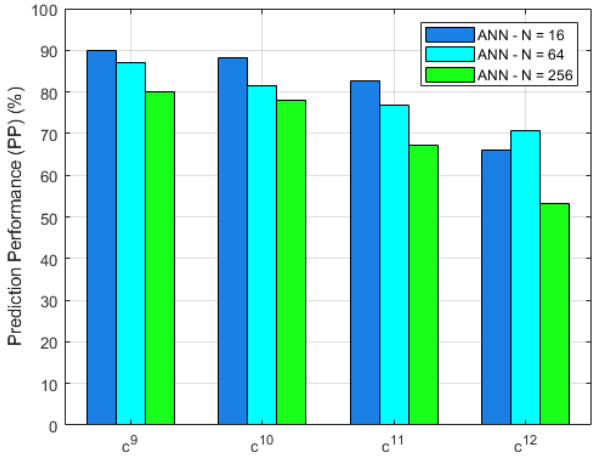

Figure 4.16: ANN prediction performance for $\Lambda = 12$, $\beta = 4$ and N = 16, 64 and 256.

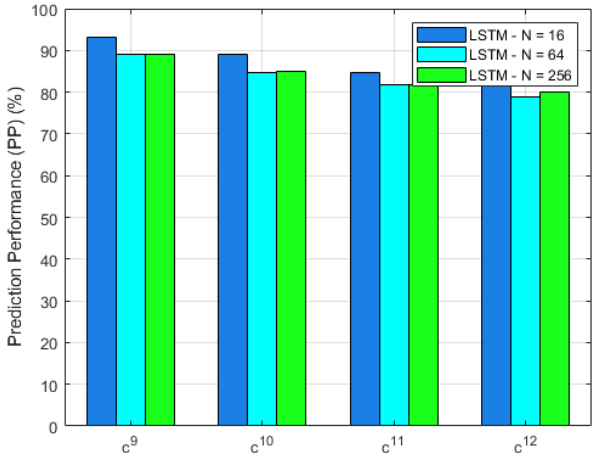

Figure 4.17: LSTM prediction performance for $\Lambda = 12$, $\beta = 4$ and N = 16, 64 and 256.

5

Conclusion

5.1 Final Considerations

This dissertation proposed two deep learning models, an ANN architecture and an LSTM architecture to predict cycling mobility in the city of Münster in Germany. The results presented in chapter 4 showed that for short-term prediction, indicating next-cell prediction ($\beta = 1$), both models had success with similar values in the prediction performance metric. The smaller grid (N = 16) had the best results for smaller sequence lengths (Λ) however, with the increase in Λ the bigger grids (N = 64 and N = 256) improved and equalized in the estimation values. This specifies that the more past cell positions knowledge the networks have, the better results they will generate regardless of the user's displacements. In the scenario of long-term prediction ($\beta = 3$, $\beta = 4$), the LSTM model outperformed the ANN model in the longer predictions. For that reason, we state that the LSTM models are the main approach for a prediction problem, not taking away the good performance from the ANN, but the computational supremacy that the LSTM displayed is clearly evident. Although, the success of this work dwells in the deep learning domain the importance of evaluating different types of parameters when choosing a dataset and the respective preprocessement of the raw data reflects in enhanced performance of these prediction algorithms.

5.2 Future Work

The work described in this dissertation focused on building a mobility prediction algorithm capable to cope with big data and being adaptable to any given environment. The success of this work stems from identifying the different mobility scenarios to customize our algorithm, efficiently preprocess big data in a way that our prediction algorithm can handle and deliver the best prediction output. Future works can include an improvement in the main components of the LSTM recurrent neural network such as the importance of the hidden units and how they affect the behavior of the model. A more successful prediction model implies a better impact in optimizing cellular networks. Thus, the next

step is to have a significant and literal impact on the mobile network by using these mobility prediction schemes as a reference to optimize processes such as routing, caching, naming, etc. By optimizing the network, our model can reach IoT applications where we will provide an enhanced personalized mobility prediction algorithm to support the development of smart applications related to our daily life.

Bibliography

[1] J. Lee et al. "Coordinated multipoint transmission and reception in LTE-advanced systems". In: vol. 50. 11. 2012, pp. 44–50. DOI: 10.1109/MCOM.2012.6353681 (cit. on p. 4).

[2] S. Parkvall, A. Furuskär, and E. Dahlman. "Evolution of LTE toward IMT-advanced". In: vol. 49. 2. 2011, pp. 84–91. DOI: 10.1109/MCOM.2011.5706315 (cit. on p. 4).

[3] A. Osseiran et al. "Scenarios for 5G mobile and wireless communications: the vision of the METIS project". In: vol. 52. 5. 2014, pp. 26–35. DOI: 10.1109/MCOM.2014.6815890 (cit. on p. 4).

[4] H. Zhang and L. Dai. "Mobility Prediction: A Survey on State-of-the-Art Schemes and Future Applications". In: vol. 7. 2019, pp. 802–822. DOI: 10.1109/ACCESS.2018.2885821 (cit. on pp. 5, 9).

[5] C. Sexton et al. "5G: Adaptable Networks Enabled by Versatile Radio Access Technologies". In: vol. 19. 2. 2017, pp. 688–720. DOI: 10.1109/COMST.2017.2652495 (cit. on p. 5).

[6] S. Lien et al. "5G New Radio: Waveform, Frame Structure, Multiple Access, and Initial Access". In: vol. 55. 6. 2017, pp. 64–71. DOI: 10.1109/MCOM.2017.1601107 (cit. on p. 5).

[7] S. A. Hoseinitabatabei et al. "The Power of Mobility Prediction in Reducing Idle-State Signaling in Cellular Systems: A Revisit to 4G Mobility Management". In: vol. 19. 5. 2020, pp. 3346–3360. DOI: 10.1109/TWC.2020.2972536 (cit. on pp. 5, 6, 8).

[8] C. Song et al. "Limits of Predictability in Human Mobility". In: vol. 327. 5968. American Association for the Advancement of Science, 2010, pp. 1018–1021. DOI: 10.1126/science.1177170. eprint: https://science.sciencemag.org/content/327/5968/1018.full.pdf. URL: https://science.sciencemag.org/content/327/5968/1018 (cit. on p. 7).

[9] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez. "Next Place Prediction Using Mobility Markov Chains". In: *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*. MPM '12. Bern, Switzerland: Association for Computing Machinery, 2012. ISBN: 9781450311632. DOI: 10.1145/2181196.2181199. URL: https://doi.org/10.1145/2181196.2181199 (cit. on p. 7).

[10] S. Qiao et al. "A Self-Adaptive Parameter Selection Trajectory Prediction Approach via Hidden Markov Models". In: vol. 16. 1. 2015, pp. 284–296. DOI: 10.1109/TITS.2014.2331758 (cit. on p. 8).

[11] L. Irio et al. "An Adaptive Learning-Based Approach for Vehicle Mobility Prediction". In: vol. 9. 2021, pp. 13671–13682. DOI: 10.1109/ACCESS.2021.3052071 (cit. on p. 8).

[12] K.-L. Yap and Y.-W. Chong. "Optimized access point selection with mobility prediction using hidden Markov Model for wireless network". In: 2017, pp. 38–42 (cit. on p. 8).

[13] C. Liu, E. Jou, and C. Lee. "Analysis and prediction of trajectories using Bayesian network". In: *2010 Sixth International Conference on Natural Computation*. Vol. 7. 2010, pp. 3808–3812. DOI: 10.1109/ICNC.2010.5583027 (cit. on p. 9).

[14] A. Nadembega, A. Hafid, and T. Taleb. "A Destination and Mobility Path Prediction Scheme for Mobile Networks". In: vol. 64. 6. 2015, pp. 2577–2590. DOI: 10.1109/TVT.2014.2345263 (cit. on p. 9).

[15] T. T. Duong and D. Q. Tran. "Mobility prediction based on collective movement behaviors in public WLANs". In: *2015 Science and Information Conference (SAI)*. 2015, pp. 1003–1010. DOI: 10.1109/SAI.2015.7237265 (cit. on p. 10).

[16] N. P. Kuruvatti et al. "Exploiting diurnal user mobility for predicting cell transitions". In: *2013 IEEE Globecom Workshops (GC Wkshps)*. 2013, pp. 293–297. DOI: 10.1109/GLOCOMW.2013.6825002 (cit. on p. 10).

[17] S. Parija et al. "Novel intelligent soft computing techniques for location prediction in mobility management". In: *2013 Students Conference on Engineering and Systems (SCES)*. 2013, pp. 1–4. DOI: 10.1109/SCES.2013.6547555 (cit. on p. 11).

[18] M. Bonola et al. "Opportunistic communication in smart city: Experimental insight with small-scale taxi fleets as data carriers". In: vol. 43. Smart Wireless Access Networks and Systems for Smart Cities. 2016, pp. 43–55. DOI: https://doi.org/10.1016/j.adhoc.2016.02.002. URL: https://www.sciencedirect.com/science/article/pii/S1570870516300257 (cit. on p. 12).

[19] J. He et al. "Delay Minimization for Data Dissemination in Large-Scale VANETs with Buses and Taxis". In: vol. 15. 8. 2016, pp. 1939–1950. DOI: 10.1109/TMC.2015.2480062 (cit. on p. 12).

[20] S. Dai, L. Li, and Z. Li. "Modeling Vehicle Interactions via Modified LSTM Models for Trajectory Prediction". In: vol. 7. 2019, pp. 38287–38296. DOI: 10.1109/ACCESS.2019.2907000 (cit. on p. 12).

[21] Y. Xing, C. Lv, and D. Cao. "Personalized Vehicle Trajectory Prediction Based on Joint Time-Series Modeling for Connected Vehicles". In: vol. 69. 2. 2020, pp. 1341–1352. DOI: 10.1109/TVT.2019.2960110 (cit. on pp. 13, 14).

[22] M. T. Asif et al. "Spatiotemporal Patterns in Large-Scale Traffic Speed Prediction". In: vol. 15. 2. 2014, pp. 794–804. DOI: 10.1109/TITS.2013.2290285 (cit. on pp. 13, 15).

[23] W. Liu and Y. Shoji. "DeepVM: RNN-Based Vehicle Mobility Prediction to Support Intelligent Vehicle Applications". In: vol. 16. 6. 2020, pp. 3997–4006. DOI: 10.1109/TII.2019.2936507 (cit. on pp. 13, 14).

[24] L. Huang, L. Lu, and W. Hua. "A Survey on Next-Cell Prediction in Cellular Networks: Schemes and Applications". In: vol. 8. 2020, pp. 201468–201485. DOI: 10.1109/ACCESS.2020.3036070 (cit. on pp. 13, 14).

[25] B. Li, H. Zhang, and H. Lu. "User mobility prediction based on Lagrange's interpolation in ultra-dense networks". In: *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2016, pp. 1–6. DOI: 10.1109/PIMRC.2016.7794984 (cit. on p. 14).

[26] S. Michaelis. "Balancing High-Load Scenarios with Next Cell Predictions and Mobility Pattern Recognition". In: Feb. 2012. DOI: 10.17877/DE290R-3340 (cit. on p. 14).

[27] M. Goodarzi et al. "Next-cell Prediction Based on Cell Sequence History and Intra-cell Trajectory". In: *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. 2019, pp. 257–263. DOI: 10.1109/ICIN.2019.8685910 (cit. on p. 14).

[28] R. Herberth et al. "Identifying Atypical Travel Patterns for Improved Medium-Term Mobility Prediction". In: vol. 21. 12. 2020, pp. 5010–5021. DOI: 10.1109/TITS.2019.2947347 (cit. on p. 15).

[29] G. Qi et al. "Analysis and Prediction of Regional Mobility Patterns of Bus Travellers Using Smart Card Data and Points of Interest Data". In: vol. 20. 4. 2019, pp. 1197–1214. DOI: 10.1109/TITS.2018.2840122 (cit. on p. 15).

[30] A. B. Adege, H. Lin, and L. Wang. "Mobility Predictions for IoT Devices Using Gated Recurrent Unit Network". In: vol. 7. 1. 2020, pp. 505–517. DOI: 10.1109/JIOT.2019.2948075 (cit. on p. 15).

[31] D. Pajarito and M. Gould. "Mapping Frictions Inhibiting Bicycle Commuting". In: vol. 7. 10. 2018. DOI: 10.3390/ijgi7100396. URL: https://www.mdpi.com/2220-9964/7/10/396 (cit. on p. 16).

[32] H. Butler et al. "The GeoJSON Format". In: draft-ietf-geojson-01. Work in Progress. Internet Engineering Task Force. URL: https://datatracker.ietf.org/doc/html/draft-ietf-geojson-01 (cit. on p. 16).

[33] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: vol. 9. 8. Nov. 1997, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf. URL: https://doi.org/10.1162/neco.1997.9.8.1735 (cit. on p. 30).

[34] L. Irio and R. Oliveira. "A comparative evaluation of probabilistic and deep learning approaches for vehicular trajectory prediction". English. In: vol. 2. Dec. 2021, pp. 140–150. DOI: 10.1109/OJVT.2021.3063125 (cit. on p. 30).

[35] T. Sousa. *Bicycles Prediction Project*. 2022. URL: https://soselius.github.io/Bicycles-Prediction/ (cit. on p. 31).