MTP-NT: A Mobile Traffic Predictor Enhanced by Neighboring and Transportation Data

Patrick Luiz de Araújo

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2023

Patrick Luiz de Araújo

MTP-NT: A Mobile Traffic Predictor Enhanced by Neighboring and Transportation Data

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Rafael Pasquini
Coorientador: Murillo Guimarães Carneiro

Uberlândia
2023

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

A663 Araújo, Patrick Luiz de, 1996-
2023 MTP-NT: A Mobile Traffic Predictor Enhanced by Neighboring Transportation Data [recurso eletrônico] / Patrick Luiz de Araújo. - 2023.

Orientador: Rafael Pasquini.
Coorientador: Murillo Guimarães Carneiro.
Dissertação (Mestrado) - Universidade Federal de Uberlândia, Pós-graduação em Ciência da Computação.
Modo de acesso: Internet.
Disponível em: http://doi.org/10.14393/ufu.di.2023.561
Inclui bibliografia.
Inclui ilustrações.

1. Computação. I. Pasquini, Rafael,1983-, (Orient.). II. Carneiro, Murillo Guimarães,1988-, (Coorient.). III. Universidade Federal de Uberlândia. Pós-graduação em Ciência da Computação. IV. Título.

CDU: 681.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074

Este trabalho é dedicado a todos os cientistas e pesquisadores do Brasil, que lutam e sonham por um futuro com mais educação, progresso e cultura.

Agradecimentos

Agradeço a Universidade Federal e todos aqueles que contribuem para sua excelência, por sempre me apoiarem em minha carreira acadêmica, em particular a Universidade Federal de Uberlândia e os professores Rafael Pasquini e Murillo, que sempre foram grandes apoiadores do meu desenvolvimento acadêmico e profissional.

"Tu és meu Brasil em toda parte
Quer na ciência ou na arte
Portentoso e altaneiro"
(Cartola)

Resumo

O desenvolvimento de técnicas que consigam realizar a previsão do tráfego de rede em uma metrópole podem alimentar aplicações *data driven*, como orquestradores de funções virtuais, chamados de Virtual Network Functions (VNF), otimizando a alocação de recursos e aumentando o número de usuários cobertos por redes móveis. Apesar de diversos estudos terem endereçado este problema, vários não consideraram a relação do tráfego em diferentes regiões da cidade e nem a informação de estações de transporte público, que podem prover informações úteis para uma melhor previsão do tráfego de rede.

Nesta pesquisa, propomos uma nova arquitetura de *deep learning* para prever o tráfego de rede usando aprendizado por representação e redes neurais recorrentes. O modelo, chamado Mobile Traffic Predictor Enhanced by Neighboring and Transportation Data (MTP-NT), tem dois principais componentes: o primeiro responsável por aprender a partir das séries temporais de uma dada região e o segundo por aprender a partir das séries temporais das regiões vizinhas e estações de transporte público. O trabalho também revisa a infraestrutura 5G baseada em especificações 3GPP abertas para explorar formas de implementar a estrutura em uma arquitetura real. Diversos experimentos foram conduzidos considerando um dataset com dados reais da cidade de Milão, assim como comparações contra técnicas estado-da-arte amplamente adotadas. Os resultados mostrados nesta pesquisa demonstram que o uso de informação de transporte público contribuem para melhorar as previsões em regiões centrais da cidade, assim como em regiões com demandas aperiódicas, tais como regiões turísticas.

Desta forma, esta pesquisa busca avaliar a performance de modelos de previsão de tráfego com o uso de dados públicos, com o intuito de validar o ganho de performance com a agregação de dados de transporte público. A agregação de dados não convencionais pode ser uma forma de adicionar informação ao modelo por meio de informações até então não exploradas no escopo desta área de pesquisa.

Palavras-chave: Redes móveis. 5G. Séries temporais. Previsão de tráfego de rede.

NFV. Deep Learning. NTMA.

MTP-NT: A Mobile Traffic Predictor Enhanced by Neighboring and Transportation Data

Patrick Luiz de Araújo

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2023

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Ciência da Computação
Av. João Naves de Ávila, 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpgfacom@ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Dissertação de Mestrado 25/2023, PPGCO				
Data:	21 de novembro de 2023	Hora de início:	08:30	Hora de encerramento:	11:05
Matrícula do Discente:	12122CCP013				
Nome do Discente:	Patrick Luiz de Araújo				
Título do Trabalho:	MTP-NT: A Mobile Traffic Predictor Enhanced by Neighboring and Transportation Data				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Sistemas de Computação				
Projeto de Pesquisa de vinculação:	-				

Reuniu-se, por videoconferência, a Banca Examinadora designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Paulo Rodolfo da Silva Leite Coelho - FACOM/UFU, Christian Esteve Rothenberg - UNICAMP, Murillo Guimarães Carneiro - FACOM/UFU (Coorientador) e Rafael Pasquini - FACOM/UFU, orientador do candidato.

Os examinadores participaram desde as seguintes localidades: Christian Esteve Rothenberg - Campinas/SP, Murillo Guimarães Carneiro, Paulo Rodolfo da Silva Leite Coelho e Rafael Pasquini - Uberlândia. O discente participou da cidade de Uberlândia/MG.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Rafael Pasquini, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Christian Esteve Rothenberg**, **Usuário Externo**, em 24/11/2023, às 16:06, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Murillo Guimarães Carneiro**, **Professor(a) do Magistério Superior**, em 24/11/2023, às 16:15, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Paulo Rodolfo da Silva Leite Coelho**, **Professor(a) do Magistério Superior**, em 27/11/2023, às 07:55, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Rafael Pasquini**, **Professor(a) do Magistério Superior**, em 30/11/2023, às 15:11, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4906667** e o código CRC **1D7CCBFD**.

Referência: Processo nº 23117.074621/2023-12 SEI nº 4906667

Abstract

The development of techniques able to forecast the mobile network traffic in a city can feed data driven applications, as VNF orchestrators, optimizing the resource allocation and increasing the capacity of mobile networks. Despite the fact that several studies have addressed this problem, many did not consider neither the traffic relationship among city regions nor information from public transport stations, which may provide useful information to better anticipate the network traffic.

In this dissertation, we propose a new deep learning architecture to forecast the network traffic using representation learning and recurrent neural networks. The framework, named MTP-NT, has two major components: the first responsible to learn from the time series of the region to be predicted, and the second one learning from the time series of both neighboring regions and public transportation stations. The work also reviews the 5G infrastructure based on open 3GPP specifications to explore ways to implement the framework in a real architecture. Several experiments were conducted over a dataset from the city of Milan, as well as comparisons against widely adopted and state-of-the-art techniques. The results shown in this work demonstrate that the usage of public transport information contribute to improve the forecasts in central areas of the city, as well as in regions with aperiodic demands, such as tourist regions.

Thus, this research seeks to evaluate the performance of traffic forecasting models using public data, in order to validate the performance gain with the aggregation of public transport data. The aggregation of unconventional data can be a way of adding information to the model through input that has not been explored in the scope of this research area.

Keywords: Mobile Networks. 5G. Time Series. Network Traffic Forecasting. NFV. Deep Learning. NTMA.

List of Figures

Figure 1 – Urban areas find throught clustering of network traffic related to their urban ecology. 24
Figure 2 – Network traffic sample from a transport region along the week. 25
Figure 3 – Network traffic concentration alog the day in Shangai. 26
Figure 4 – Network traffic in different regions of the city 27
Figure 5 – Autocorrelation in uplink and downlink data. 28
Figure 6 – Model architecture with Local Stacked AutoEncoders (LSAE) and Global Stacked AutoEncoders (GSAE) arrangements. 29
Figure 7 – Prediction performance improvement in Downlink (top) and Uplink (bottom) scenarios. 30
Figure 8 – Decomposition of network traffic with different traffic characteristics. (a) In-tower traffic dominant, from a residential area; (b) consistent inter-tower dominance along the day, in a shopping mall; (c) Inter-tower traffic dominance at certain times, from a transit station. 31
Figure 9 – Mathematical neuron. 34
Figure 10 – Deep Learning architecture. 35
Figure 11 – Long short-term memory (LSTM) cell architecture with each equation of the mathematical formalization pointed out. 35
Figure 12 – LSTM layer with a chain of connected cells. 37
Figure 13 – Conventional network based on proprietary hardware architecture. Source: the authors. 40
Figure 14 – Implementation option of NFV architecture to the network in Figure 13. Source: the authors. 41
Figure 15 – End-to-end information flow to MTP-NT proposed architecture. In black, the normal requisitions and the dashed purple line shows the publish-subscribe like communications. 43
Figure 16 – Absolute traffic in 5 regions in Milan: Duomo, Bocconi, Navigli, Mesiano and Bosco. 46

Figure 17 – Mapping of public transport in the city of Milan. In blue, green and red the metro, tram and bus stops, respectively. 47
Figure 18 – Proposed framework architecture, detailing the two branches used for prediction. 50
Figure 19 – Model architecture, highlighting the branches and the different stages of model processing. 53
Figure 20 – Compiled from average results for the 64 regions investigated, with neighborhood data degree ranging from 1 to 5, without using data from the public transport system. 57
Figure 21 – Results for the 64 investigated regions, with neighbor degree varying between 1 and 5, using data from the public transport systems. 58
Figure 22 – Histogram of tests in region 607, showing the original Call Detail Records (CDR) data and the predictions in logaritmic scale. In blue the real data and in orange the predictions with transport information. 59
Figure 23 – Results for region 8169. In blue, the real network usage CDF and in orange the model predictions CDF. 59
Figure 24 – Execution time for each approach. 61
Figure 25 – Results for different Holt-Winters (HW) implementations. 64
Figure 26 – Plot for San Siro/Giuseppe Meazza region predictions considering transport hubs. In blue the real network usage, in orange the model predictions with the aperiodic peak highlighted. 65

List of Tables

Table 1 – Spatial correlation of a arrange of 7 Base Stations (BSs) 28
Table 2 – Comparison between different network traffic prediction studies. 32
Table 3 – Comparison between different network traffic prediction studies. 32
Table 4 – Original data from the dataframe, showing multiple samples with the same Square id and Time Interval (in timestamp) to register calls during the measurements (using the Country Code feature). 45
Table 5 – Sample data after the preprocessing process. 48
Table 6 – Number of regions and data samples in a 24 hour interval with increasing neighborhoods. 50
Table 7 – Normalized Mean Absolute Error (NMAE) in tests with 10-minute and 1-hour observations, varying the usage of transport hubs. 60
Table 8 – NMAE among different benchmarking techniques in Distributed, Core and Event tests. 62

Acronyms list

3GPP 3rd Generation Partnership Project

ACF AutoCorrelation Function

AI Artificial Intelligence

ARIMA AutoRegressive Moving Average

BS Base Station

BSs Base Stations

CDR Call Detail Records

CN Core Network

DFT Discrete Fourier Transform

eNBs Evolved Node B

ETSI European Telecommunications Standards Institute

GDPR General Data Protection Regulation

GNN Graph Neural Network

GSAE Global Stacked AutoEncoders

HW Holt-Winters

IDS Intrusion Detection System

IDSs Intrusion Detection Systems

ISP Internet Service Provider

LGPD Lei Geral de Proteção de Dados

LSAE Local Stacked AutoEncoders

LSTM Long short-term memory

ML Machine Learning

MTP-NT Mobile Traffic Predictor Enhanced by Neighboring and Transportation Data

MLP Multi Layer Perceptron

NFV Network Function Virtualization

NMAE Normalized Mean Absolute Error

NN Neural Network

NSSF Network Slice Selection Functions

NTMA Network Traffic Monitoring and Analysis

NWDAF Network Data Analytics Function

PCF Policy Control Function

QoE Quality of Experience

QoS Quality of Service

RAN Radio Access Network

ReLU Rectified Linear Unit

RNN Recurrent Neural Networks

SLP Single Layer Perceptron

SMS Short Message Service

SVR Support Vector Regressor

TSF Time Series Function

WMMSE Weighted Minimum Mean-Square Error

VNF Virtual Network Functions

VoLTE Voice over LTE

Contents

1 INTRODUCTION 17

2 RELATED WORK 23
2.1 Network traffic characterization 24
2.2 Network traffic prediction 26

3 THEORETICAL FUNDAMENTATION 33
3.1 Neural networks development and training 33
3.2 LSTM 34
3.3 Feature selection 36

4 FRAMEWORK AND PREPROCESSING 39
4.1 The predictive model in the 5G infrastructure 40
4.2 Data flow in MTP-NT 42
4.3 Dataset 44
4.4 Mathematical formalization of dataset preprocessing 47

5 MTP-NT AS A OPEN SOURCE FRAMEWORK 49
5.1 Mathematical formalization of MTP-NT operations 49
5.2 MTP-NT's framework architecture 51

6 EXPERIMENTAL RESULTS 55
6.1 Experimental setup 55
6.2 Error evaluation 56
6.3 Execution time evaluation 60
6.4 Benchmarking 61
6.5 Concluding remarks 64

7 FINAL CONSIDERATIONS AND FUTURE WORK 67

BIBLIOGRAPHY **71**

APPENDIX **77**

APPENDIX A – README OF THE SOURCE CODE **79**

I hereby certify that I have obtained all legal permissions from the owner(s) of each third-party copyrighted matter included in my thesis, and that their permissions allow availability such as being deposited in public digital libraries.

Patrick Luiz de Araújo

Chapter 1

Introduction

At the end of 2028 there will be near 5 billion of 5G nodes (ERICSSON, 2022). All connected nodes will generate an average of 100 exabytes of data per quarter, generating a massive demand of traffic on mobile networks (mainly 3G, 4G, 5G and Wi-Fi). In this scenario, it will be of crucial importance for mobile network providers being able to allocate the maximum amount of users and optimizing the network operability leveraging on Network Function Virtualization (NFV) (Sun et al., 2019). To deal with that, Machine Learning (ML) and other predictive tools can be used to improve resource allocation and respond quickly to changes in network traffic.

These tools have a main role in this scenario, so resource allocation (on the user perspective) and network slicing (on the telecommunications operator perspective) can achieve better results. There is also other ML usage and research in mobile networks, as techniques to improve energy efficiency of the network (Niu, 2011), approaches to optimize resource sharing to the gNodeB (gNB) in 5G networks and data driven procedures for deployment planning of base stations (Lee et al., 2014).

In addition to the goal of allocating an increasing number of devices with an increasing demand for bandwidth, 5G networks have additional goals to achieve, as near $1ms$ latency, low energy consumption and almost 100% coverage, for example(Agiwal; Roy; Saxena, 2016). To achieve these objectives, caching in network edges, cloud computing based infrastructure and many other techniques can be used to improve the network efficiency and, in almost all of them, Artificial Intelligence (AI) can be used to boost those techniques and increase the impact on the network.

5G networks, in particular, are designed to be managed through VNF of the Core Network (CN) in a NFV topology - that will be executed in the cloud and/or edge, allowing resource management in an unified way (Alawe et al., 2018). Besides network management, VNF will allow the creation of flexible networks under demand, scaling according to the amount of resource requested or even to the network traffic itself. However, this flexibility can create imbalanced networks, with BSs with different capabilities and behaviors, causing the overload of the network even before it reaches its maximum capacity

(Gotzner; Rathgeber, 1998).

As conventional algorithms in network management were designed for static networks, the imbalance of traffic loads among heterogenous Base Station (BS)s needs to be addressed. Here, AI models have an important role, with a mathematical modelling that normally rely on historical data (Wang et al., 2017b; Wang et al., 2017a; Chen et al., 2015).

AI solutions offers some advantages when used in network problems. As shown in (Sun et al., 2017) a well designed network for power allocation can be less complex than a Weighted Minimum Mean-Square Error (WMMSE) algorithm. Those tecniques generally learn robust patterns and avoid past faults with a better overall performance (Sun et al., 2019). However, some points can limit the usage of ML solutions. To cite a few:

- ❏ Do the problem have enough data so a model can be trained and evaluated in?
- ❏ All the pertinent information related to the problem is being correctly used? All the information used in the model are pertinent and have a causality relation with the problem?
- ❏ The response time of the final architecture allows it to be used in real time?

To review successful works in AI and 5G, this dissertation make a bibliographical review from both areas, with a theoretical review of ML and the basis of the new generation of mobile networks and its new paradigm, focusing on the topics of infrastructure virtualization and the use of data and predictive models in the CN. The subarea of the junction between the two areas explored in this dissertation is the branch of developing network traffic prediction models in mobile networks, where ML models are developed with a focus on assisting the CN in its internal processes by providing network traffic predictions.

In this dissertation we design a new deep learning architecture to forecast the network traffic using representation learning and Recurrent Neural Networks (RNN). The proposed framework is called Mobile Traffic Predictor Enhanced by Neighboring and Transportation Data (MTP-NT) and uses a Recurrent Neural Networks (RNN) along with a representation encoding composed by two main components: one component process information regarding both neighborhood and public transportation data, which assumes the concept that information provided by the network traffic analysis among city regions and, as well as, from public transportation stations may provide useful information to better predict network traffic; and the other component processes the past time series of the region of interest to be predicted, which is motivated by the concept that network usage present high periodicity, and, as consequence, high self correlation in some specific time intervals (Wang et al., 2017a).

Considering such a predictive context, the main contributions from our investigations in this dissertation include:

- ❏ The development of an easily implementable framework to network traffic predictions;
- ❏ Reproducibility of our findings given the fact that all the data is available under Creative Commons Attribution 4.0 International License and the code used in this work is documented at (de Araújo; Pasquini; Murillo Guimarães Carneiro, 2022) and its readme can be seen in appendix A;
- ❏ An implementation based on a General Data Protection Regulation (GDPR) compliant dataset;
- ❏ Best overall performance among the evaluated time series prediction techniques;
- ❏ Validation of performance with tests with real world data;
- ❏ Tests on a highly dense public event on Giuseppe Meazza/San Ciro stadium, in Milan, validating MTP-NT performance on extreme scenarios of network consumption.
- ❏ A method to use public transportation data into network traffic modeling;
- ❏ Testing and usage of feature selection techniques to ensure that just information that adds value to the overall performance are used;

The integration of public transportation data into network traffic modeling represents a promising avenue in the aggregation of useful data to the network traffic forecasting modeling. This data not only enhances the accuracy of traffic models but also shows a new kind of information that can be explored in projects of this nature.

The development of a replicable framework in open source license is important as not only aligns with ethical and collaborative values but also fosters a robust and transparent research ecosystem. Researchers can leverage these techniques to drive innovation, build upon existing work, and contribute to the collective advancement of knowledge. This is also important to allow the solution to be implemented and tested in real use cases in the industry.

According to (ALLIANCE, 2022), 80% of the sites in a city carry 20% of the traffic; 50% of them carry 5% of the traffic; and 50% of the data is consumed in less than $0,35\%$ of the area, which creates high stress zonal areas in the network infrastructure (mostly in central areas). With a high demand for a disproportionately distributed network in large centers and the deployment of increasingly flexible and virtualized networks, big cities have a wide range of opportunities to benefit from network usage predictions.

The simplest case is to allocate more precisely the network core resources, as some regions of a city might have a lower network demand rather than central regions in peak areas, for example. Also, as MTP-NT uses public transportation data, in bigger cities, with more granular and robust public transportation infrastructure, the framework could perform better predictions compared to cities with smaller public transport infrastructures.

The overall idea of MTP-NT generalizes to different mobile network technologies. However, given intrinsic properties of 5G related to Network Function Virtualization, it creates a favorable scenario to deploy our proposal. Therefore, in this dissertation, we present our proposal as an exercise of instantiating MTP-NT based on 5G technologies. We focus on the conceptual definition of key functions from 5G architecture required to support our proposal.

The dissertation is ordered as it follows: Chapter 2 presents related work about network traffic characterization and network traffic forecasting. The first is a branch of research that aims to find patterns on the network consumption in different areas of the city and moments of the day. The second tries to predicts the network demand for further usage on the mobile network infrastructure.

Chapter 3 gives a theoretical overview of Machine Learning (ML) techniques, focusing on the main topics related to MTP-NT. Chapter 4 revisits some of the research in the area of network traffic characterization and network traffic prediction, as well as how the mobile network backend evolved in the fifth-generation of networks by means of Network Function Virtualization (NFV) and cloud. It is also shown how MTP-NT could be applied to the 5G infrastructure in details. The dataset is described, presenting the data used, the format and any transformations that were made.

In Chapter 5, we delve into a precise mathematical formalization of MTP-NT. This chapter not only elucidates the mathematical principles underpinning the framework but also unveils the architectural components of the technique. It provides a comprehensive overview of how the neural network is structured and details the specific data employed within each segment (or branch) as it will be later explained.

Chapter 6 shows the results from the experiments by comparing MTP-NT to related works. Different tests where made to ensure the overall performance of the proposed framework, making previsions in Base Stations (BSs) with different network usage patterns to ensure that MTP-NT has a good performance regardless the scenario. The main result of the work, regardless its good overall performance of prediction, is the test of the framework in a high scale event near Giuseppe Meazza/San Ciro stadium, where the predictions were able to keep up with a high aperiodic network traffic behavior, as will be shown later. Different tests varying the amount of data were also performed so that the impact of increasing the volume of data used can be measured from the point of view of training time and quality of predictions.

Chapter 7 contains final considerations of the work, recapitulating all the development of the work and impact on the current scenario of telecommunications, as well as showing possible next steps of the research.

Chapter 2

Related work

As explained by (Xu et al., 2017), the network traffic characterization can be very useful for both network operators and the government. The modelling of network traffics can be used by the Internet Service Provider (ISP) to customize load balancing and strategies to specific towers/regions (related to specific traffic patterns) and increase the optimization of the resource usages as well as Quality of Service (QoS)/Quality of Experience (QoE). To government and other organizations related to urban planning (as transport companies, for example) the traffic analysis can be a useful source of land usage and human economic activites.

There are two main groups of studies related to network traffic analysis: network traffic characterization and prediction. Investigations that deal with the characterization of network traffic aim to analyze metrics and mathematical characteristics of network usage (Xu et al., 2017; Wang et al., 2015; Gotzner; Rathgeber, 1998)

Other studies focus on the development of mathematical models to predict network traffic and, according to the review in (Boutaba et al., 2018), the problem can be modelled in two different ways: traffic prediction as a pure Time Series Function (TSF) problem, where the network traffic is manipulated as a time series function, and as a non-TSF problem, where other methods are used, such as frequency-domain analysis.

TSF approaches rely on past observations of the data and, to make a viable mathematical model to predict a time series, it's needed previous past samples to train, validate and test the model. Unfortunately, in complex telecommunications scenarios and high speed data links, it might be technically difficult to conduct measurements at the required speed and granularity.

In this way, some studies try to predict network traffic by other methods and features. Even though these non-TSF approaches had shown, in general, less accuracy than TSF approaches, the predictions can be done with more complex models and ensamble learning. (Boutaba et al., 2018) offers further details and clarification of the main research in the area.

2.1 Network traffic characterization

In (Xu et al., 2017), for example, different regions of a city are grouped based on network usage patterns, dividing them into residential, transport, office, entertainment and comprehensive areas. The used dataset consists of 9,600 BSs serving 150,000 users, with data from August 1st to August 31st 2014.

The data was condensed in 10-minute samples (the same way as the data used in this dissertation (Barlacchi et al., 2015)) and, after a data exploration and preprocessing, a clustering algorithm based on the time series of the network traffic in each region was performed to find the best arrangement of clusters. The optimal number of clusters was found using Davies-Bouldin algorithm (Maulik; Bandyopadhyay, 2002).

The resulting 5 clusters, modelled by the network traffic, can be related to their urban ecology, based initially on human labelling of a few regions and generalized to the rest, resulting in the residential, transport, office, entertainment and comprehensive areas as seen in Figure 1.

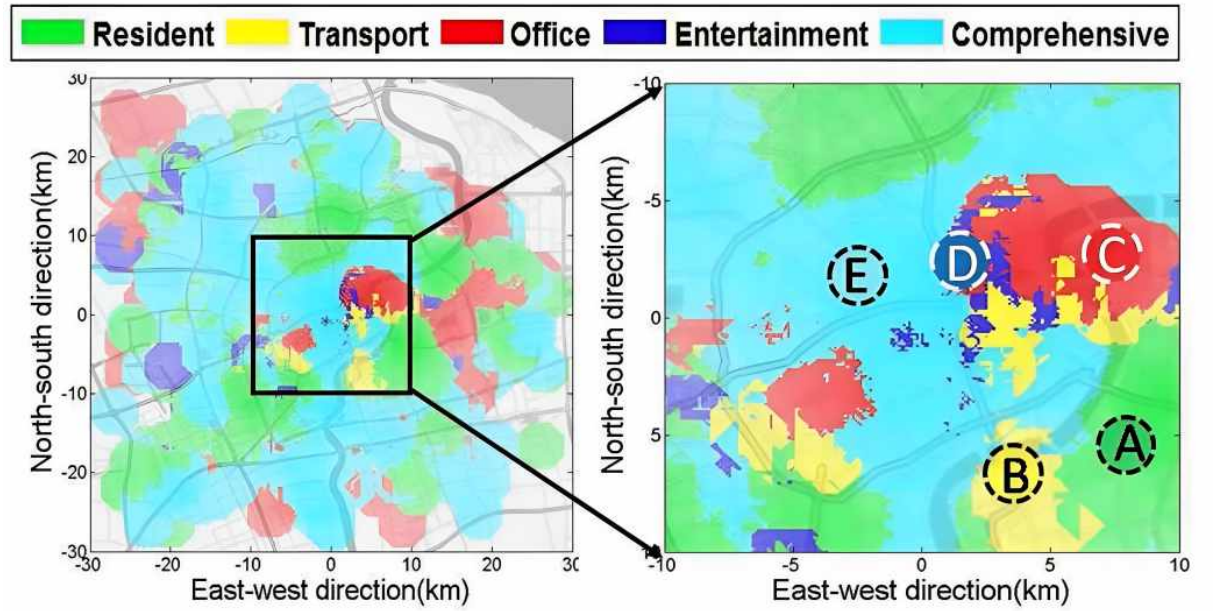


Figure 1 – Urban areas find throught clustering of network traffic related to their urban ecology.

Source: (Xu et al., 2017).

Some characteristics of the clusters are also explored in time domain, which are:

1. Weekday-Weekend Traffic Amount Ratio: ratio of weekday and weekend traffic, which shows how much of the network traffic is concentrated in those different periods of the week. Residential areas, for example, have an ratio of 1, while office

areas have a ratio near to two, related to the fact that people don't tend to go to these areas and use the network (related to the work days);

2. Peak-Valley Features: the ratio of the maximum and the minimum traffic registered during the analysis. In transport areas, for example, this peak is 133.33, which shows the high discrepancy of network usage in rush hours and other periods of the day in these regions during the week, as can be seen in Figure 2;

3. Time of Traffic Peak and Valley: the analysis of the time of the day of maximum and minimum network usage can also be a good source of information, showing the time characteristics of the demand.

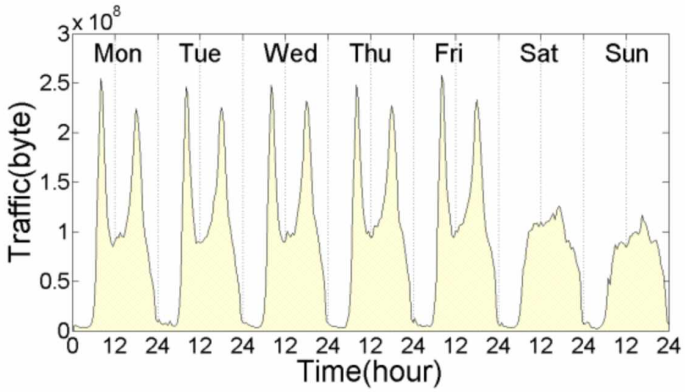


Figure 2 – Network traffic sample from a transport region along the week.

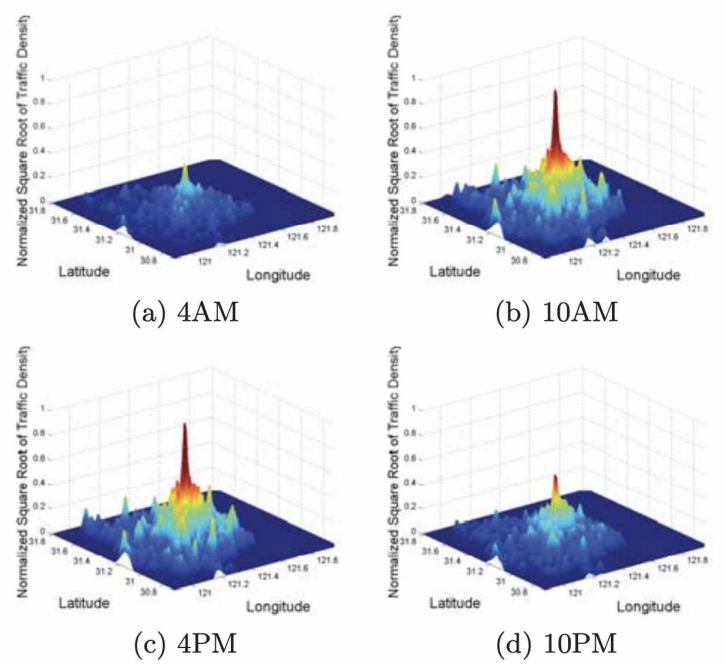
Source: (Xu et al., 2017).

The time series are also explored in the frequency domain through Discrete Fourier Transform (DFT), exploring the contributions related to week, day and halfday periodicity. After this characterization, the network traffic is modelled through the frequency components explored, showing good results.

(Wang et al., 2015) explored an arrange of 380,000 BSs in Shanghai from August 1 to August 31, 2014. The data also have 10 minute samples for each Base Station (BS) during the period with recordings of data communication, of which the device's ID (anonymized), starting and ending time, BS ID, location and traffic volume. The total amount of data reaches 1.96 billion entries with a total size of 2.8 PB (92 TB per day, 7 GB per BS on average).

The work find that the mobile traffic follow a trimodal distribution composed by compound-exponential, power-law and exponential distribution and that could be possible to have a R-square of 99% describing the network traffic by means of a trimodal distribution. Another conclusion was that, in urban regions, the traffic is non-homogeneous, being centered in specific time and area.

Besides the study around the trimodal distribution as a way to describe the network traffic, (Wang et al., 2015) also analyzes the spatial and temporal distribution of the network traffic, highlighting that the concentration of the network consumption (in both temporal and spatial terms) as can be seen in Figure 3.



(a) 4AM (b) 10AM

(c) 4PM (d) 10PM

Figure 3 – Network traffic concentration alog the day in Shangai.

Source: (Wang et al., 2015).

As seen in (Xu et al., 2017), the work also discusses the difference in the network traffic demand related to the urban ecology, as can be seen in Figure 4. All the inhomogeneity results into a extremely insufficient utilization of network resources.

(Gotzner; Rathgeber, 1998) analyzed the traffic in the city of Berlin between May 1996 and August 1997 and found that the traffic was concentrated in the city center, generating congestion in these regions in peak hours, while in other points of the city the network resources were not fully used. This makes the infrastructure reaches the maximum capacity before all processing power could be used.

Despite the mobile network generation during the study was not the same as today, the data consumption is much more related with the displacement pattern and behavior, making the conclusions still valid.

2.2 Network traffic prediction

In the first documented usage of Neural Network (NN) to traffic forecasting, a Multi Layer Perceptron (MLP) network was used back in 1993, supported by mathematical

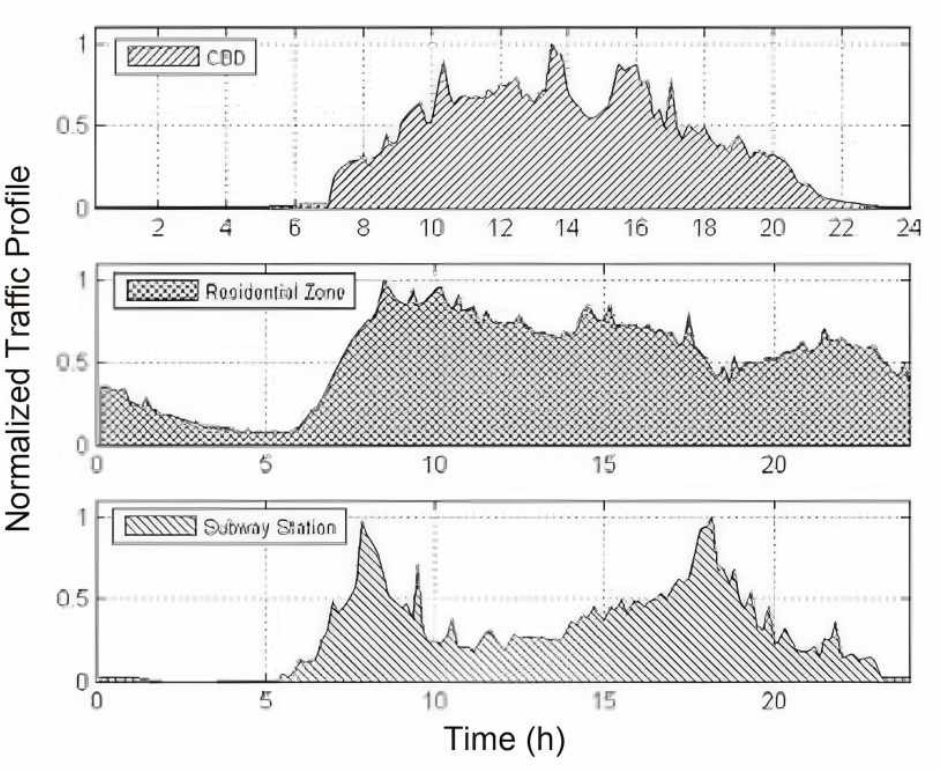


Figure 4 – Network traffic in different regions of the city

Source: (Wang et al., 2015).

proofs from previous years that presume the use of Single Layer Perceptron (SLP) and MLP to traffic prediction (Cybenko, 1989; Hornik, 1991; Funahashi, 1989). Since then, much has evolved and recent studies focus on the development of mathematical models to forecast network traffic (Wang et al., 2017a; Wang et al., 2017b; Hanyu Yang et al., 2021). In these studies, which are related to the investigation conducted on this dissertation, data modeling techniques are applied in an attempt to predict the network traffic in the city infrastructure.

Since then, the Artificial Intelligence (AI) area has evolved and become quite popular, and in popular programming languages, such as Python, libraries package the main components used in neural networks, such as TensorFlow and Keras (KERAS, 2023). MTP-NT is based mainly in Long short-term memory (LSTM) layers (Hochreiter; Schmidhuber, 1997), dropout layers (Baldi; Sadowski, 2014) and conventional dense layers implemented in Python using Keras. In Section 5 there is a mathematical approach to describe each of these components.

In (Wang et al., 2017a), a China Mobile database with samples of $2,844$ BSs in the city of Suzhou, with a coverage area of $6,500km^2$, between May and October of 2015 was used. The total coverage area of each BS has been simplified to a grid of $500m \times 500m$.

The study shows that, despite being widely used, techniques such as Support Vector Regressor (SVR) and AutoRegressive Moving Average (ARIMA) does not captures rapid

variation process as it relies basically in mean values of the past series data, as well as does not captures spatial dependency of the data (such as correlation of neighboring BSs).

To validate that, a preliminary data analysis was performed to explore the data dependency in both temporal and spatial domains. Related to the autocorrelation, the AutoCorrelation Function (ACF) was used to discover the data dependency in the temporal domain, and the results can be seen in Figure 5. A high downlink and uplink traffic correlation can be observed in time lags of one or multiple of 24 (hours), showing a daily pattern.

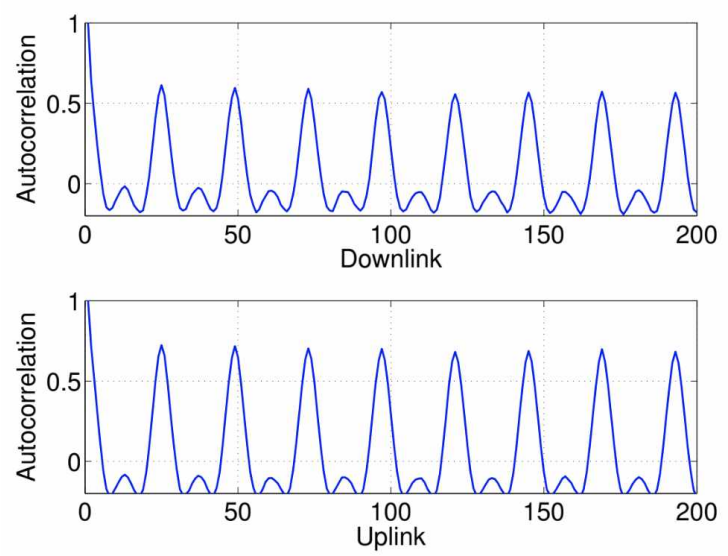


Figure 5 – Autocorrelation in uplink and downlink data.

Source: (Wang et al., 2017a).

The spatial correlation was also explored using the covariance and the standard deviation. The results can be seen in Table 1. Each of the 7 cells are subsequently located on the east side of the previous one and the upper part of the table shows the uplink, while the lower triangular part is for downlink data.

Table 1 – Spatial correlation of a arrange of 7 BSs

	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6	Cell 7
Cell 1	1.000	0.167	0.435	0.130	0.040	0.341	0.307
Cell 2	0.396	1.000	0.338	0.129	0.084	0.310	0.222
Cell 3	0.345	0.541	1.000	0.159	0.162	0.697	0.536
Cell 4	0.437	0.439	0.458	1.000	0.104	0.131	0.114
Cell 5	0.360	0.471	0.492	0.508	1.000	0.163	0.080
Cell 6	0.286	0.491	0.550	0.432	0.535	1.000	0.603
Cell 7	0.284	0.506	0.526	0.459	0.535	0.577	1.000

Source: (Wang et al., 2017a)

With all these considerations, the model developed aims to capture both historical and spatial data. The architecture relies on a RNN forecast combined with Global Stacked AutoEncoders (GSAE) (that process all the stations of the grid) and Local Stacked AutoEncoders (LSAE), which achieved promising results.

As seen in Figure 6, each BS has its data, combined with its neighborhood time series, processed through the LSAE and a LSTM, where each data arrange have its own encoder and LSTM combination, to make the local correlations. Furthermore, the GSAE receives all data arranges and the output is added in each individual arrange before the LSTM to embed in the model the global information (as it processes all the data of the grid). The idea of multiple branches in the model was also explored in the development of MTP-NT as will be later explained. The hybrid model, with global and local branches, leads to some benefits, as better representation and support for parallel training, for example.

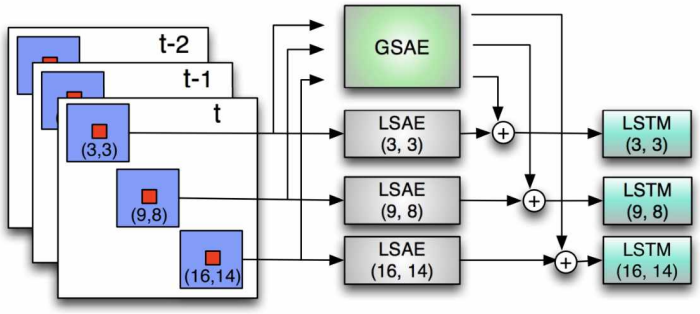


Figure 6 – Model architecture with LSAE and GSAE arrangements.

Source: (Wang et al., 2017a).

Besides the model complexity, the work reaches a good overall performance, seen in an evaluation that also observed the performance increase with the increase of hidden layers in LSAE. This prediction performance improvement depending on the number of hidden layers can be seen in Figure 7 and a similar approach was also performed in MTP-NT to explore the possibility of less complex models, best suited to scenarios of reduced computational resources or faster predictions scenarios.

As seen, (Wang et al., 2017a) adopts some approaches in the model development that were used in this dissertation, such as the concept of neighborhood region of interest and LSTM cells to make better predictions based on previous data. The similarity between the resulting grid arrange from the simplification of coverage regions and the database used in this dissertation also show similarities to each other.

In (Wang et al., 2017b) the authors investigated an urban area with 5929 towers and about to 1.5 million users by decomposing the traffic into in-tower (static users who have not performed a handoff) and inter-tower (users that came from neighboring towers). It is shown that inter-tower traffic can account for up to 90% of the entire data traffic at a transportation hub.

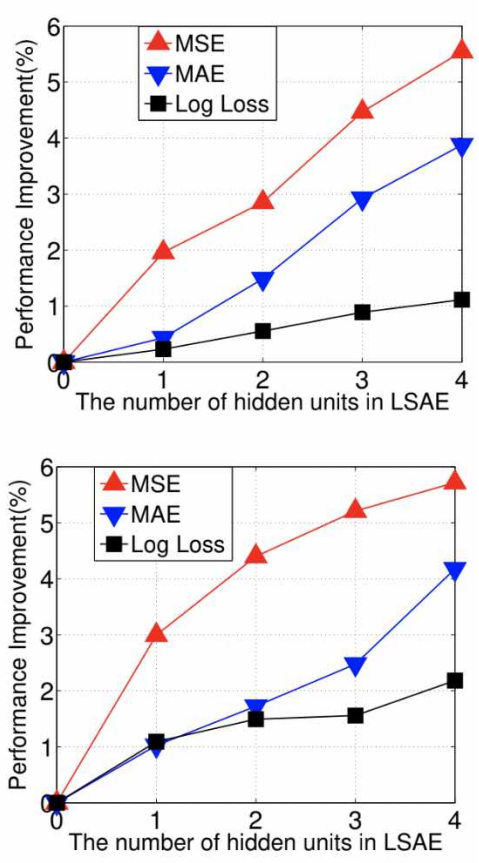


Figure 7 – Prediction performance improvement in Downlink (top) and Uplink (bottom) scenarios.

Source: (Wang et al., 2017a).

Futhermore, the evolution of public transport led to a more efficient urban mobility as fast travel within a metropolis, what ends up increasing the correlation between the BSs that serve public transport hubs and physically distant towers. However, most of the traffic prediction solutions fails to capture these long-distance spatial dependency of the traffic. This concept was explored in this dissertation, where, as will be explained later, the public transport data were directly inserted in the modeling of the predictive model.

(Wang et al., 2017b) has one of the most detailed datasets when compared to the other works explored in this dissertation, where each entry in the database contains the user ID, flow create time, connected BS as well as uplink and downlink traffic aggregated in half hour samples. Unfortunately, this means that the implementation of this project or a simillar approach might not be possible in regions that follows stricts privacy regulations, such as GDPR and the brazillian Lei Geral de Proteção de Dados (LGPD), as the information might led to the personal identification of identity, location and other information of every network user.

The traffic of each BS was decompose on in-tower and inter-tower traffic, where the first one is related to stationery users and the second to high mobility users (i.e. those

who comes from another BS and performs a handoff). Figure 8 shows the decomposition of the network traffic from three different BSs.

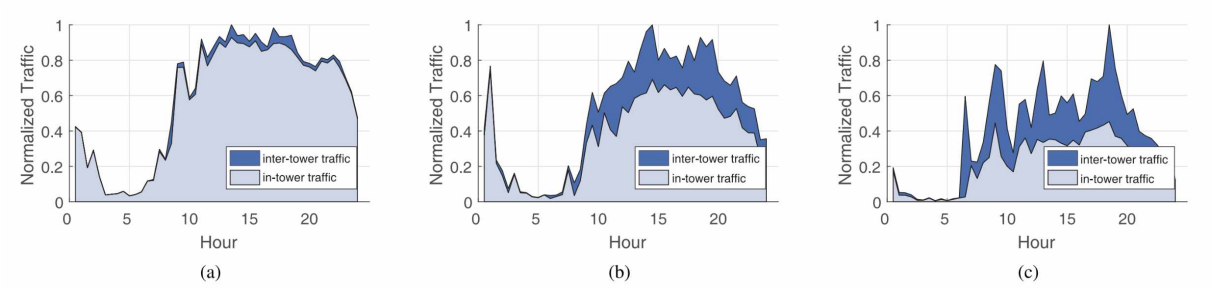


Figure 8 – Decomposition of network traffic with different traffic characteristics. (a) In-tower traffic dominant, from a residential area; (b) consistent inter-tower dominance along the day, in a shopping mall; (c) Inter-tower traffic dominance at certain times, from a transit station.

Source: (Wang et al., 2017b).

A graph representation of the data is used, representing a composition of the BS and the spatial dependency. A Graph Neural Network (GNN) based model was developed to predict the future network traffic in the towers and results show that even in a unbalanced database the model has a good overall performance.

In order to contemplate the greatest number of possible scenarios, including abnormalities in network consumption, this work also explored seasonal events (those that repeat themselves periodically), trends (events that provide a continuous increase or decrease over time) and, mainly, mobility-independent events. This last set of scenarios aims to validate the performance of the model in events such as political speeches, shows, sport games, traffic jams, etc and were also considered on the tests of MTP-NT.

In (Hanyu Yang et al., 2021) a network composed by an ARIMA and a neural network were proposed, in which the first architecture was used to extract linear components and the second for non-linear components. Furthermore, the architecture is trained using the Simulated Annealing (SA), similar to the cooling process of metallurgy. This training technique presents promising results when compared to traditional time series forecasting methods.

What has been observed in the area of traffic forecasting is that most of the researches were not so easily replicable, as the base software codes are not publicly available and the implementation of such proposals are not available for external use. The most related works to this proposal can be seen on Table 2, where is shown that most of the research have little information about the dataset and all of them do not provide the codes to simplify the reproducibility of the projects.

Each work considers different parameters, as cell neighborhood or the relation between

Table 2 – Comparison between different network traffic prediction studies.

Ref.	Method	Dataset	Sourcecode avalaible
(Wang et al., 2017a)	Autoencoders	Private	No
(Wang et al., 2017b)	Graph Neural Networks	Private	No
(Sciancalepore et al., 2017)	HoltWinters	No information	No
(Alawe et al., 2018)	Deep Learning	Private	No
(Hanyu Yang et al., 2021)	ARIMA and Neural Network	Available	No

Source: the author

the stations through network handoffs. What is sought is, in short, that relevant information correlated to the target is identified in order to allow the construction of a predictive model by using data as less as possible.

Nonetheless, each described technique has an element that increases considerably the training time and the execution of the routines, such as the construction of many encoders or the usage of lots of additional data (from neighboring BSs), making the training process costly and the model more complex.

In summary, the productions considered during the development of this dissertation - besides characterization studies (Wang et al., 2015; Gotzner; Rathgeber, 1998; Boutaba et al., 2018) - can be divided into 7 major topics, directly related to the methodology used in this work:

Table 3 – Comparison between different network traffic prediction studies.

Ref.	LSTM	Time Series	Neighborhood	Spatial modelling	Events	Traditional models
(Wang et al., 2017a)	x		x	x		
(Wang et al., 2017b)		x	x	x	x	
(Hanyu Yang et al., 2021)		x				
(Sciancalepore et al., 2017)						x
(Alawe et al., 2018)						x

Chapter 3

Theoretical fundamentation

Almost all related work of network traffic prediction as well as MTP-NT itself relies on ML techniques. This area gained attention in many research areas throughout the last years because of its capability to solve complex problems without explicit programming, learning useful information from historical data. The proliferation of data in various scientific disciplines has reached unprecedented levels, facilitated by advancements in data collection and storage technologies, boosting ML widely adoption in both academic and industry scenarios.

The vast majority of ML implementations are based on "conventional" neural networks and RNN, mainly used in time series modelling because of its capacity to deal with sequential data (also called time series). All of these will be later explained in this chapter.

3.1 Neural networks development and training

NN, a cornerstone of modern ML, are computational models that mimic the neural connections and information processing capabilities of the human brain. Neurons are at the core of neural networks, which serve as the basic building blocks.

The neuron has three main components: the inputs X, the weights W and the activation function θ, resulting in the output $\hat{y}$ as seen in Figure 9. Both inputs and weights are matrices.

First, the inputs X are multiplied by the weights W and then added to the bias b, as seen in Equation 1. The function of the bias is to avoid problems during training when the inputs are zero.

$$Z = X \times W + b \tag{1}$$

After that, the result is applied in a activation function θ as seen in Equation 2. Activation functions in neural networks are mathematical functions that are applied to

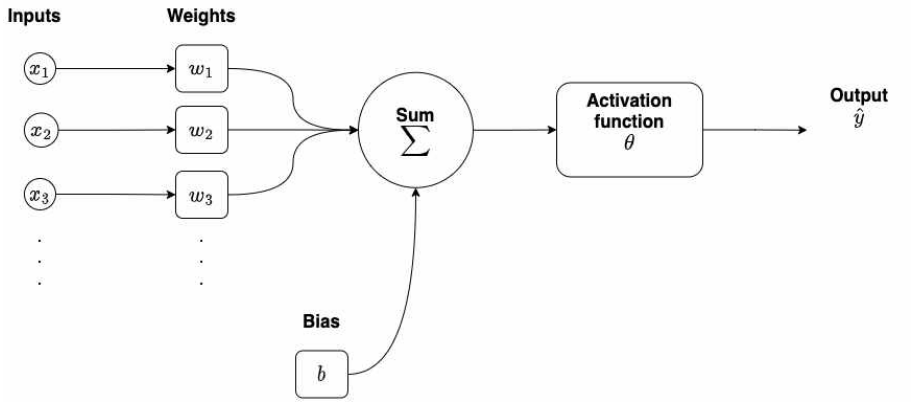


Figure 9 – Mathematical neuron.

Source: the author

the weighted sum of inputs at each neuron to introduce non-linearity into the neuron's response, allowing neural networks to capture intricate patterns and make more expressive predictions. Commonly used activation functions include the sigmoid function, which maps inputs to a range between 0 and 1; the hyperbolic tangent function, mapping inputs to a range between -1 and 1. The Rectified Linear Unit (ReLU) has gained prominence due to its computational efficiency and ability to mitigate the vanishing gradient problem (in training).

$$\hat{Y} = \theta(Z) \tag{2}$$

These neurons (also called cells) are organized in layers in a NN, where groups of neurons passes its outputs to the next layer. If a NN have one or more layers between the input and output layers, it is called Deep Neural Network, as it allows even deeper connections between the neurons as seen in Figure 10.

From all parameters in the NN, just the weights are capable to be changed, as the inputs are variables external to the network. In neural networks, we say that the weights are trainable parameters.

To adapt the NN to solve a problem (making the predictions $\hat{y}$ be as close as possible to the real value y), there is an iterative process called backpropagation. The method takes a neural networks output error (the difference between the predictions and the real values) and propagates this error backwards through the network adapting the weights of the neurons to minimize the error (or cost function, as it is called in the AI field.

3.2 LSTM

LSTM layers are composed of many LSTM cells, used in time series and image modeling, being used in many implementations as language modeling, handwriting recognition,

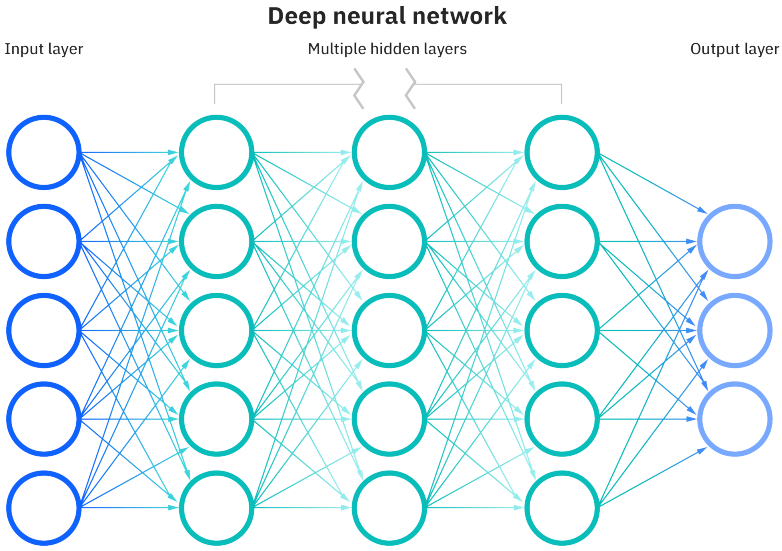


Figure 10 – Deep Learning architecture.

Source: (IBM, 2023).

translation tasks, and speech synthesis, e.g. (Graves; Schmidhuber, 2005; Sutskever; Vinyals; Le, 2014; Graves et al., 2006; Zen et al., 2013). These networks are capable of learning long-term dependencies using the "cell state", a mechanism that maintains its state over time and allows to use past states as input in current and future predictions. The LSTM layer is composed of many cells, as seen in Figure 11 (with red flags linked to the equations of the mathematical formalization), interconnected as seen in Figure 12.

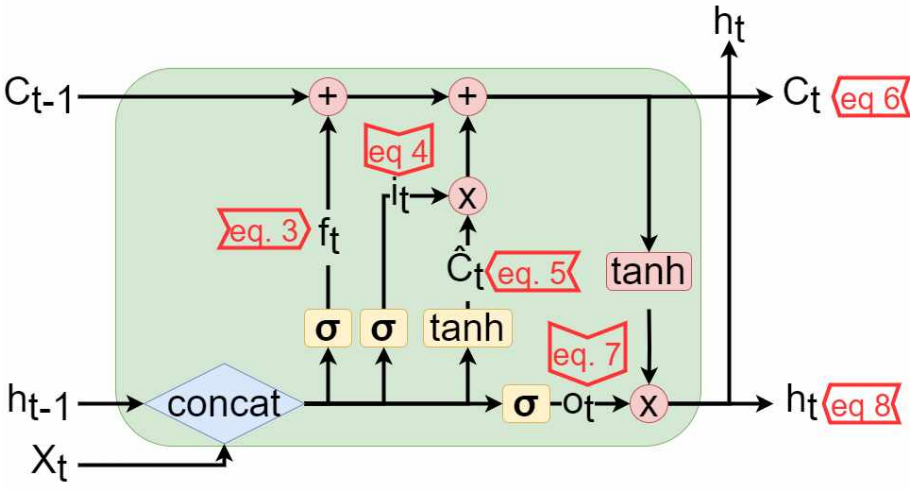


Figure 11 – LSTM cell architecture with each equation of the mathematical formalization pointed out.

Source: the author

Despite the LSTM layer has a single input from the previous layer and a single output

to the next one, the LSTM cell has three inputs, C_{t-1}, the $t-1$ cell state (coming from the previous LSTM cell h_{t-1}, the previous output, and x_t, the actual input data. From the layer perspective, x_t represents the input of the previous layer and h_t the output.

The first step in the LSTM cell is to decide whether the previous state will be preserved or not in the "forget gate" with the output in f_t, where W_f are the weights of the actual network, (h_{t-1}, x_t) are the concatenated x_t and h_{t-1} and b_f is the bias, as seen in Equation 3.

$$f_t = \sigma\left[W_f \cdot (h_{t-1}, x_t) + b_f\right] \tag{3}$$

The next step is to define the set of previous data which is going to be stored in the cell state. The operation has 2 steps, where i_t, called "input layer gate", decides which values are going to be updated as seen in Equation 4.

$$i_t = \sigma\left[W_i \cdot (h_{t-1}, x_t) + b_i\right] \tag{4}$$

Next, a vector of context candidates $\check{C}_t$ is generated with a hyperbolic tangent operation as seen in Equation 5.

$$\check{C}_t = tanh\left[W_c \cdot (h_{t-1}, x_t) + b_c\right] \tag{5}$$

After all these calculations, the final cell state C_t is calculated as seen in Equation 6.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \check{C}_t \tag{6}$$

On the end, the output h_t is calculated through Equation 7 and Equation 8.

$$o_t = \sigma\left[W_o \cdot (h_{t-1}, x_t) + b_o\right] \tag{7}$$

$$h_t = o_t \cdot tanh\left(C_t\right) \tag{8}$$

All operations inside the LSTM cell are neural networks operations, with updatable weights through backpropagation and the LSTM cells forms a chain of structures in a layer, as seen in Figure 12. A more detailed and mathematical formalization can be seen in (Hochreiter; Schmidhuber, 1997; OLAH, 2015).

3.3 Feature selection

Another concept adopted in MTP-NT is the feature selection, which involves the identification of the most salient features that significantly contribute to accurate classification outcomes. The primary objective of feature selection is to reduce the dimensionality of the feature space while preserving or enhancing classification performance. By selecting

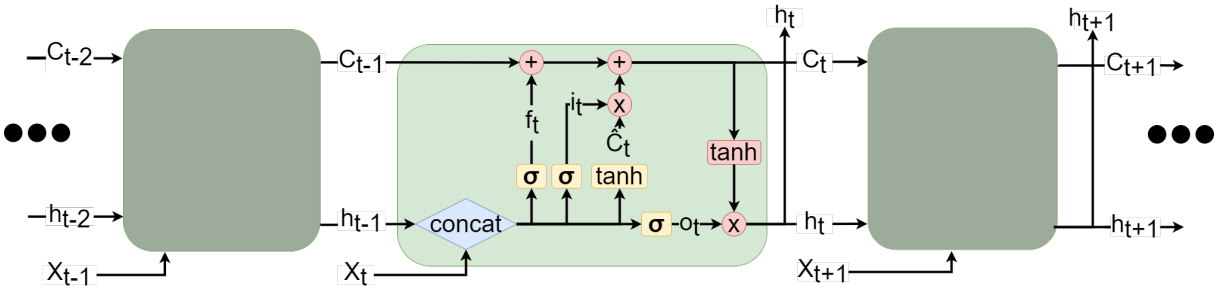


Figure 12 – LSTM layer with a chain of connected cells.

Source: the author

a subset of relevant features, the computational cost is lessened, and the interpretability of the classification model is improved.

The feature selection was mainly used to identify the temporal components most pertinent to predicting network usage in MTP-NT. Three feature selection techniques were tested: Pearson correlation, f-value coefficient and a simple distance-based algorithm (that picks the components from a given maximum distance).

The Pearson correlation coefficient is a number between -1 and 1 that measures the strength and the direction of the relationship between two variables. With this information, the n-biggest correlated variables (in module) or all variables with a correlation coefficient above a certain threshold can be selected to compose the model input. The f-value was used over Scikit Learn (SCIKITLEARN, 2022), a popular data science library in Python.

Chapter 4

Framework and preprocessing

As seen in (Gotzner; Rathgeber, 1998) along the day in big cities the network usage tends to stay concentrated in specific regions during peak hours, while other regions use to have a lower network usage at that time. This behavior was also explored in (Xu et al., 2017), where different regions of the city are classified according to their general composition (residential areas, central areas and so on) and, consequently, segregating according to their network utilization patterns.

Both observed behaviors might lead to underutilization of core network resources in certain areas that, in big cities, results in higher infrastructure costs. Residential areas, for example, tend to have a higher network usage by the beginning and the end of the day (moments when people tend to be going out or coming home from work) and along the day the network usage tends to be lower, while in commercial areas the logic used to be reverse (as the flow of people tend to be more common in business hours).

As new mobile network protocols, such as 5G, are based on infrastructure virtualization, the resources could be entirely deployed and dynamically allocated on cloud services, allowing to dynamically adjust the infrastructure capabilities (linked to the amount of computational resources available to each service).

With the possibility of leveraging on different computational resources available on the network infrastructure, the unique issue to minimize costs while achieving agreed QoS/QoE metrics is the capability to make the right decisions in terms of resource allocation and to ensure that every part of the system have the necessary resources to satisfy the network demand, even in highly dynamic scenarios. MTP-NT can provide this intelligence in strategic areas of the city, as central regions and points with high usage of computational resources, resulting in potential savings to the network operators. As mentioned earlier, MTP-NT is a ML model that helps the scheduler to better optimize the resource allocation process by providing a network traffic prediction.

4.1 The predictive model in the 5G infrastructure

In previous network infrastructures, some specific functions were performed by middleboxes, hardware-based applications as firewall, Intrusion Detection System (IDS), proxy, encryption, data monitoring and other services. These services were usually deployed on proprietary hardware, placed at fixed locations and needed specialized personal for deployment and maintenance (Bari et al., 2016), as seen in Figure 13.

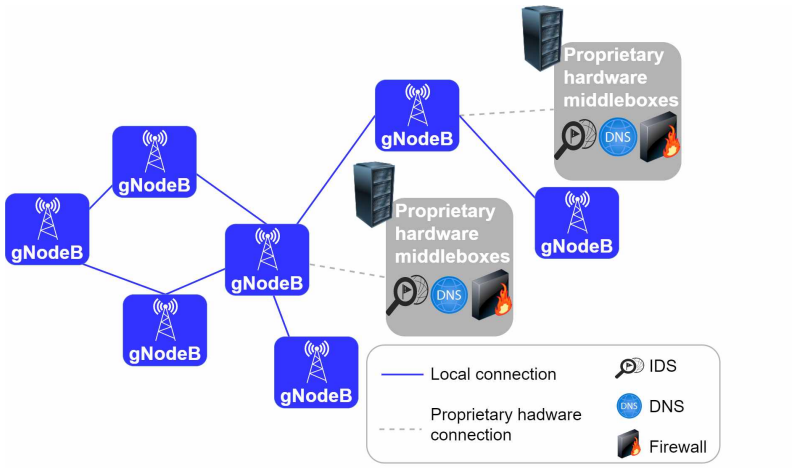


Figure 13 – Conventional network based on proprietary hardware architecture. Source: the authors.

These middleboxes are static hardware that perform single tasks, not allowing new functionality and subjecting the telecommunications operator to short deployment and replacement cycles to keep up with new demands and technologies. A suitable option to improve this architecture, as seen in new networks like 5G, is the use of NFV, an approach where those services provided in the network become software based middleboxes, called Virtual Network Functions (VNF), typically as Virtual Machines or Containers. Instead of relying on proprietary hardware in the classical middleboxes approach, these functions are running on both cloud and local servers (on the NFV architecture) and could be dynamically provisioned based on the network needs.

(Herrera; Botero, 2016) make a brief explanation of the main advantages and usages of NFV in the industry and, according to European Telecommunications Standards Institute (ETSI) (ETSI, 2013), some advantages of network virtualization that stand out are:

1. NFV as a service: a NFV can be provided as a service by a network operator similar to cloud computing services (Rankothge et al., 2015);

2. Virtualization of Core Network (CN) and BSs (Basta et al., 2014);

3. Virtualization of the home environment: installation of new equipment and on-site technical support can be less frequent (Bronstein; Shraga, 2014);

4. Virtualization of CDNs (Mangili; Martignon; Capone, 2014; Kim; Lee, 2014).

The main barrier of this new approach is the overall performance, especially in middlebox chains. Some traffic could flow through various middleboxes based on their needs, e.g., a proxy request that need to pass through firewall, Intrusion Detection Systems (IDSs) and the proxy service itself. However, the literature shows that the NFV can achieve almost a hardware-based performance (Martins et al., 2014; Hwang, 2014).

Based on the NFV architecture, middleboxes could rely on different local servers or even remote ones in the cloud, as far as they could achieve the QoS and QoE requirements and can also have their resources dynamically optimized, mainly oriented to save financial and computational resources. This new approach allows also that conventional middleboxes could be used with cloud as well as general purpose local hardware, allowing a mix architecture to attend many necessities. An architectural example can be seen in Figure 14, where al BSs are connected to each other and to local middleboxes as well as to cloud services thought internet (gray dashed lines).

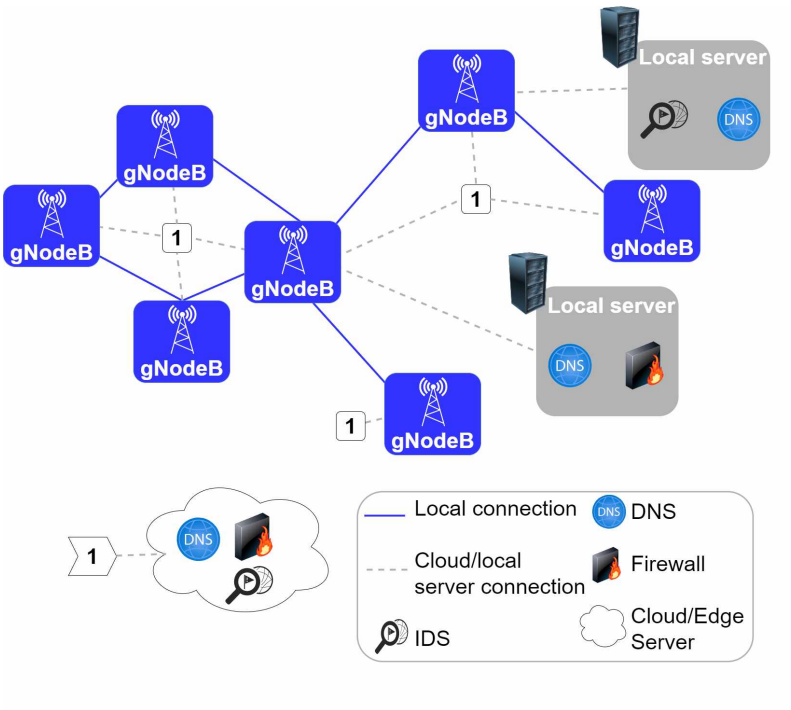


Figure 14 – Implementation option of NFV architecture to the network in Figure 13. Source: the authors.

The optimization of NFV services are done by the orchestrator, which find the optimal scenario to the network services based on service metrics and also on Network Traffic Monitoring and Analysis (NTMA). NTMA are VNFs that, based on historical data, try to predict a wide range of network metrics and could be in 8 different categories (Boutaba R., 2018), including QoS and QoE management and traffic prediction, aligned with the focus of the architecture proposed in this dissertation.

As explained in (D'Alconzo et al., 2019), the NTMA have a special problem with the high volume of data associated to this task, along with speed (needed to both gauge and process the data). These monitoring tasks pose as big data problems, which have in common many of the "5V's" challenges for these kind of architectures, of which it is possible to highlight:

1. Volume: in Milan itself, Telefonica has $10,000$ base stations to provide mobile network services. The huge volume of data generated in a big city demands data treatment and processing capable of dealing with the large volume of information generated in an acceptable speed;

2. Velocity: to supply the scheduler with more granular data, NTMA systems have to work in forecasts with an increasing frequency, so that the scheduler can work on an increasingly accurate management of resources.

To feed these traffic prediction applications, a data storage system, capable of maintaining measurements of the network, is necessary to save previous metrics, to receive data from current measurements and to supply NTMAs with requested data. To make this possible, NoSQL solutions are the best alternatives, as they suffer less performance penalties with large datasets (Han et al., 2011; D'Alconzo et al., 2019) and have some popular open source options, such as Cassandra and HBase.

4.2 Data flow in MTP-NT

To supply all network traffic information in the database of MTP-NT, flow collectors (also referred as network exporters and collectors), located at the Radio Access Network (RAN) layer, collect the necessary metrics. In (Barlacchi et al., 2015), the data is aggregated in squared regions in the city, so one part of the grid do not represents a respective tower, but a area of the city covered by one or more BSs, as the exact locations of the Evolved Node B (eNBs) are not disclosed to preserve the coverage strategies of the network provider. Despite this anonymization, regions can be aggregated based on the tower that covers them and the predictions made by the framework can be used in this same aggregation to predict future traffic, now aggregated based on towers. These predictions can be made available through a publish-subscribe messaging system, such as Kafka for example. The flow collector can be a Network Data Analytics Function (NWDAF), an network analytics provider as a logical function specified at 3rd Generation Partnership Project (3GPP) SA2 TS 23.682 (3GPP, 2022) that collects information from the network and notifies any interested VNF instance about all analytics subscribed over the data management service.

The NWDAF architecture is quite versatile, as may interface with a Policy Control Function (PCF) layer (a platform to govern the policy management based on network parameters and that implements slice-based policies) through a N23 interface, and Network Slice Selection Functions (NSSF) through N34 interface (Chouman; Manias; Shami, 2022). The NWDAF can provide useful information to allow automate network issue resolution, while predictive analytics can be used to predict those network issues in the future.

After the data ingestion and prediction, the network traffic prediction from MTP-NT can be remapped from the region-based aggregation to a BS-based view (where the network traffic is compiled by BS and not by region) to allow the predictions to be more alligned with the scheduler task of fairly distribution of radiotemporal resources by BS and processing power (on premise and cloud) by BS.

Therefore, the data can be both stored into a database for further use and send it to other VNF, as NSSF and other components that can take advantage from the predictions. In order to send the predictions directly to other core network services, the NWDAF can be used. The NWDAF architecture is quite versatile, as may interface with a PCF entity (a platform to govern the policy management based on network parameters and that implements slice-based policies), and interface with NSSF (3GPP, 2022) over a publisher-subscriber like protocol for internal use in 5G core (3GPP, 2022)). In this way both the resource allocator of radio resources, from the mobile network layer and the resource allocator of the computational resources in the processing layer can use these predictions to make better decisions based on future information and not only in previous measurements or conventional algorithms.

The complete MTP-NT information flow can be seen in Figure 15 and the complete step by step information flow, from the RAN layer to the NTMA model, is described as follows:

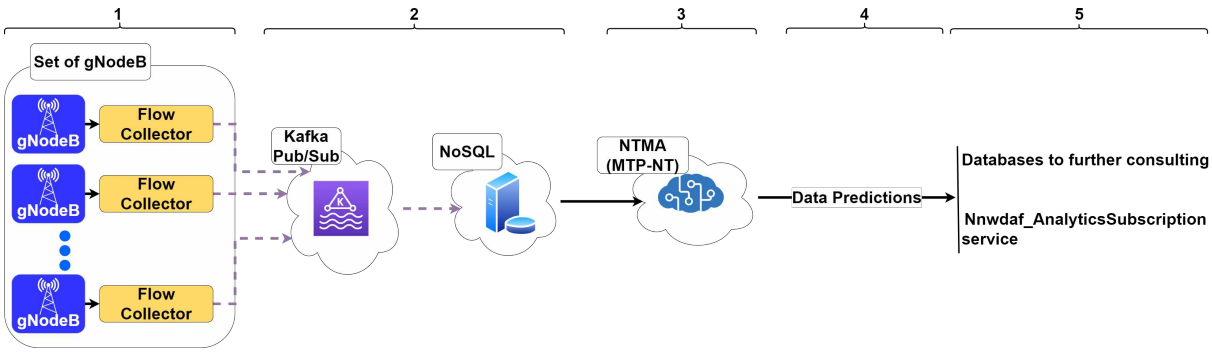


Figure 15 – End-to-end information flow to MTP-NT proposed architecture. In black, the normal requisitions and the dashed purple line shows the publish-subscribe like communications.

1. Flow collectors collect and compile network traffic data from gNodeBs;

2. The data reaches the database, on a VNF, from the flow collectors over a publish-subscribe architecture;

3. The requested data from the database reaches the NTMA model predictors, also on VNFs;

4. The model processes the data and generates the traffic predictions;

5. The traffic predictions go to a external database to further use, and are made available through `Nnwdaf_AnalyticsSubscription` service (3GPP, 2022).

With many options of implementation available, it can be benefic to rely on open source techonology, as this paradigm promotes equitable access to software solutions, reducing the reliance on costly licenses and proprietary lock-ins. There are many NoSQL open source technologies, as the aforementioned Cassandra and Redis, many publisher subscriber open source options and the general compliance with 3GPP architecture are interesting options to guarantee an architecture "open first".

4.3 Dataset

The database used in this study contains 7 groups of data: Grid (Telecom Italia), Social Pulse (Spazio Dati, DEIB), Telecommunications (Telecom Italia), Precipitations (Metereotrentino, ARPA), Weather (ARPA), Electricity (SET Distribuizione SPA) and News (Citynews) for the cities of Milan and Trento, in Italy. This work used the Telecommunications dataset in Milan, which contains the mobile network traffic between November 1st, 2013 and December 31st, 2013 on the $10,000$ zonal regions listed in the city, containing one traffic log at every 10 minutes in each point. The sum of the regions results in a grid of 100×100 over the city of Milan, that covers all the metropolitan area of the city and each region is, approximately, $0.06km^2$.

In this database, the network traffic is measured in Call Detail Records (CDR), and each CDR is generated every time a user initiates or ends a network connection. For a given connection, an additional CDR is generated every 15 additional minutes of connection or if the user transfers more than 5MB over the internet. The CDR are also used in Short Message Service (SMS) and calls, but this information is not relevant in this work.

It is important to note that this database aggregates network traffic across regions and not across towers. This means that, after the entire stage of collecting and pre-processing the original traffic, the network traffic is aggregated in regular regions of $0.06km^2$ and this could have been done to preserve the coverage strategy of the city.

To protect the real dimension of infrastructure capabilities, the numbers of daily operation of the company and to guarantee a GDPR compliant data set, each field of the

database is multiplied by an anonymization constant k defined by Telecom Italia. Futhermore, there is an anonymization constant to every parameter of the database, in order that the proportion of the values is kept, at the same time that the real values are protected and the task of traceback a specific user is impossible. A sample of the telecommunications data can be seen in Table 4, where the data is: *Square_id* (identifying the region of the mesh), *Time_Interval* (with the timestamp of the sample), Country code (a column used to identify the other side of a phone call, in case of a call in the sample), SMS-in and SMS-out activity (registering if there is any input or output of message), Call-in and Call-out activity (recording data input and output via phone call, respectively) and Internet traffic activity, registering internet connection related CDR).

Table 4 – Original data from the dataframe, showing multiple samples with the same Square id and Time Interval (in timestamp) to register calls during the measurements (using the Country Code feature).

Square id	Time Interval	Country code	SMS-in activity	SMS-out activity	Call-in activity	Call-out activity	Internet traffic activity
1	1383606E+6	0	1.7873E-3	null	null	null	null
1	1383606E+6	33	null	null	null	null	2.6137E-2
1	1383606E+6	39	8.8512E-2	1.4195E-1	1.0804E-1	2.73E-2	9.2032
10	1383606E+6	33	null	null	null	null	2.8653E-2
10	1383606E+6	39	6.7480E-2	1.0631E-1	5.9175E-2	1.0174E-2	5.7891

Source: the author

As can be seen in Table 4, there may be more than one register for a single region (`Square id`) in the same time interval. It occurs when there are more than one sender/receiver of information, so that it can be possible to register traffic from different phones with different country codes. The *null* samples represents that there is no traffic for the given BS and time interval, but they are all handled during the processing.

In Figure 16 it is shown a sample of the connection's numbers in 5 regions (Duomo, Bocconi, Navigli, Mesiano and Bosco). The X axis represents time throughout the week, and the Y axis represents the absolute number of connections. Unfortunately, it is not clear from (Barlacchi et al., 2015) which is the first day of the week in these plots, but the general idea of weekly seasonality and network usage proportion of more and less crowded areas can be seen.

In the upper graph, it is possible to observe that each region has its own traffic pattern. The surroundings of the university (Bocconi) have a drop in traffic on the weekends, while Duomo and Navigli present more intense traffic, result of a greater flow of people in those touristic regions. In the bottom graph the disparity between the traffic amplitude in Duomo (touristic region) and the other regions stands out, making the traffic in Bosco become almost imperceptible, as it is a less crowded region. Besides having their differences, which come from the characteristics of each region, the time series are noticeably correlated, presenting certain periodicity and being a good indicative that previous samples can be used in the future traffic forecasting.

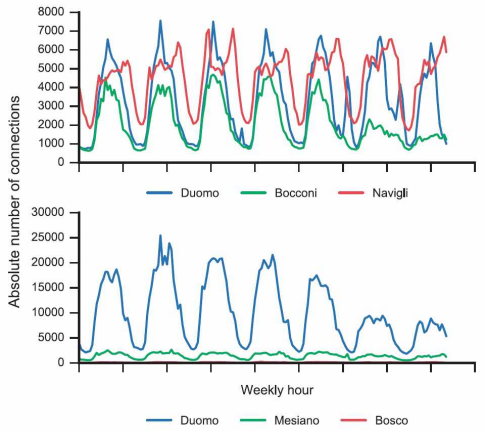


Figure 16 – Absolute traffic in 5 regions in Milan: Duomo, Bocconi, Navigli, Mesiano and Bosco.

Source: (Barlacchi et al., 2015)

Sudden changes in network usage can make the predictions very inaccurate, as network behavior can be very irregular (D'Alconzo et al., 2019). This concept can also be extended to the human behavior in general, as the way people move around the city slightly changes over the time. Despite the changes, patterns can be (usually) identified and models could be developed as seen in (Pirozmand et al., 2014).

(Wang et al., 2017b) made an analysis of the network traffic dissecting the volume of data between in-tower (static users who have not performed a handoff) and inter-tower (users that came from neighboring towers). In this work, it is discussed that the evolution of public transport led to more efficient and faster travel within a metropolis, what ends up increasing the correlation between the BSs that serves public transport hubs and physically distant towers. However, most of the traffic prediction solutions fails to capture these long-distance spatial dependency of the traffic.

Despite being a strong indicative of being a solution to the long-distance correlation of network traffic, the literature review carried out in this work did not find any application of public transport data in improving the performance of network traffic predictive models.

The conclusion is also feasible when analyzed in terms of urban ecology as proposed in (Wang et al., 2017a). Cities and metropolitan regions tend to expand in area with the creation of new residential neighborhoods within the city limits. However, as business and administrative centers tend to remain in the same place, increasingly faster and more efficient means of transport are built and improved to ensure the smooth functioning of urban logistics.

In addition to being an indicative of the future flow in certain regions, the transport hubs also can be used to infer non-periodic events that were not covered in the training, such as concerts, political speeches, sport events and other activities that were not regular (Wang et al., 2017b), but may cause a high impact on the network.

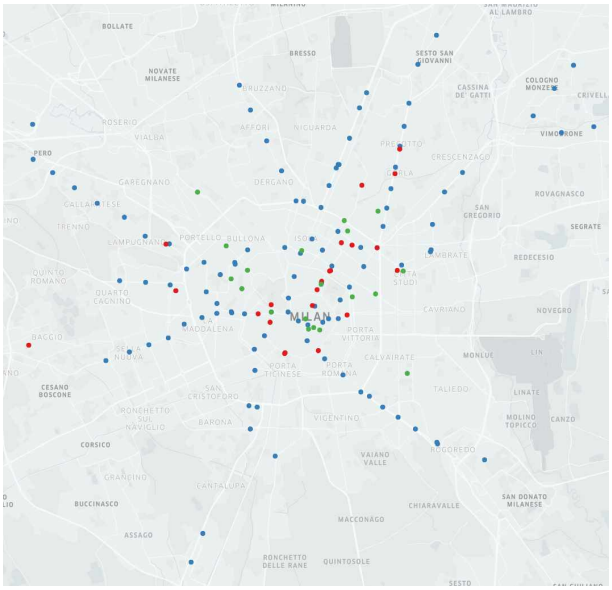


Figure 17 – Mapping of public transport in the city of Milan. In blue, green and red the metro, tram and bus stops, respectively.

Source: the author

According to (CITIVATIS, 2021), the main public transports in Milan are metro, tram and bus. The mapping of their main stations can be seen in Figure 17 and the metro points were mapped from the data of *Azienda Trasporti Milanesi* (ATM), the subway company of Milan, available at (ATM, 2021). The tram and bus stops were collected using Google Maps API. To use these data in the framework, the transportation hubs are located in terms of the network coverage map (relating a hub to a `square id` of the city's regions) and the final transportation data is a list of `square ids` that have a tram, metro or bus station.

This dissertation aims to explore the CDR generated by network connections, as network usage have become the main purpose of mobile devices, increasing its importance when compared to other types of traffic. However, the other groups of data provided could be explored in further projects as a way to make more precise predictions. As will be explained later, MTP-NT has a neural network architecture versatile enough to allow new sets of information to be inserted through new branches.

4.4 Mathematical formalization of dataset preprocessing

As mentioned before, the original data is structured in such a way that there are more than one registry per cell per time period and, as this work just uses the network traffic data, all data with the same `square id` and time interval were combined and the internet

traffic activity was added. Thus, there are $10,000$ network traffic logs (one for each region) and, for each log, there are 8928 traffic samples, 1 for each 10-minute interval and, with that, the atomic unit of the base station becomes the sum of CDR for 10 minutes in a given region of the city. The data from Table 4, after the processing, became the data seen in Table 5. After all processing, there is one sample to each *square_id* and timestamp combination and the sequence of Internet traffic activity CDR to one *square_id* ordered by the timestamp is the time series to that designated region, formalizated as $x^{m,n}(t)$ to the region (m, n) in the timestamp t.

Table 5 – Sample data after the preprocessing process.

Square id	Time Interval	Country code	SMS-in activity	SMS-out activity	Call-in activity	Call-out activity	Internet traffic activity
1	1383606E+6	72	9.0299E-2	1.4195E-1	1.0804E-1	2.73E-1	9.2294
10	1383606E+6	72	6.7480E-2	1.0631E-1	5.9175E-2	1.0174E-2	5.8178

Source: the author

The mathematical formalization of a sample from a region x in a time period t can be seen as $x^{m,n}(t)$, where t is the timestamp that goes from $13832604E+5$ to $13886166E+5$ and m, n are between 0 and 100 and represents the coordinates of the region. Also, as mentioned earlier, all *null* samples have been correctly handled and are no longer present in the data.

As mentioned previously, MTP-NT only processes the internet traffic activity, which was mainly motivated by the fact that network usage is much more significant in the volume of data processed by the network infrastructure than the other types of traffic presented. However, the preprocessing algorithm compiles all traffic sources, which could be useful in future work that wishes to consider other network traffic modes, especially considering that telephone calls and even SMS are starting to be treated as packets. (as Voice over LTE (VoLTE), for example).

After all preprocessing, the data is ready to be ingested by MTP-NT, with one sample to each region in each timestamp. In the following chapter it will be explained how this data is handled internally by the framework.

Chapter 5

MTP-NT as a open source framework

5.1 Mathematical formalization of MTP-NT operations

To model the degree of proximity of regions in the construction of MTP-NT the concept of neighborhood was used and, as seen in Figure 18, the degree of neighborhood between two regions implies the distance between them, formalized by Moore neighborhood concept. The Moore neighborhood is a concept used in cellular automata theory and, in a square grid, a point m', n' can be considered in d neighborhood of a point m, n if it satisfies the following Equation 9:

$$|m - m'| \leq d, |n - n'| \leq d \tag{9}$$

In summary, degree 1 neighbors are at Moore neighborhood of $d = 1$ and degree 2 neighbors are at Moore neighborhood of $d = 2$ from the central region, for example. For a region $x^{m,n}$, the group of neighbors N with a degree d can be seen at Equation 10.

$$N(x^{m,n})_d = \{x^{m',n'} \mid |m - m'| \leq d, |n - n'| \leq d \forall m, n, m', n' \in \{0, 1, ..., 99\}\} \tag{10}$$

The increment of data is a tradeoff between the size of the model (and subsequently the computational cost involved into the training and execution) and the quality of the predictions (as it will be discussed further). MTP-NT experiments was done from 1 to 5 neighbors and, as seen in Table 6, the increment of regions causes a considerable increment in the number of regions and the number of data points.

As there is a model for each region and each model considers a closed set of neighborhoods and transport hubs, MTP-NT can be an option for grids that cover urban centers of any size. Naturally, larger regions (and with more regions) will need more models to have all predictions needed. However, because they are individualized models, it is possible to distribute and parallelize the execution of MTP-NT.

Table 6 – Number of regions and data samples in a 24 hour interval with increasing neighborhoods.

Neighborhoods	**1**	**2**	**3**	**4**	**5**
Total regions	9	25	49	81	121
Samples in 24 hours	1,296	3,600	7,056	11,664	17,424

Source: the author

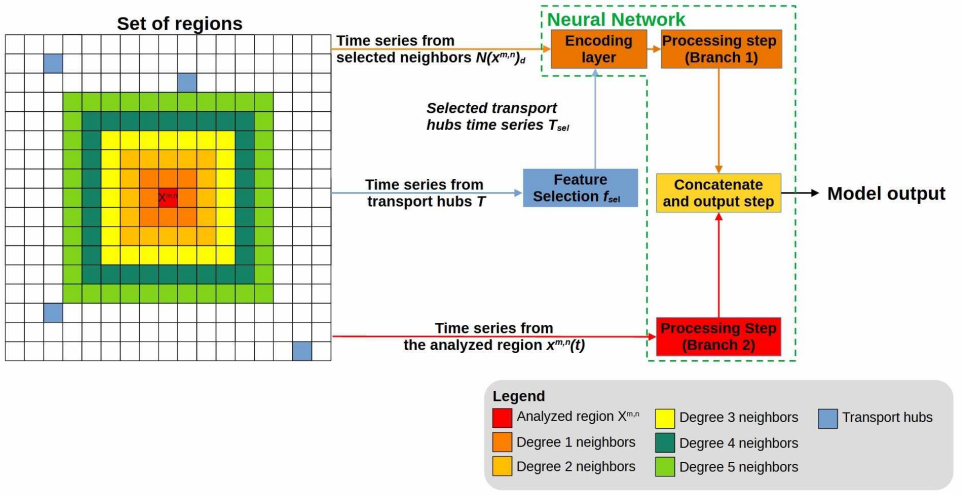


Figure 18 – Proposed framework architecture, detailing the two branches used for prediction.

Source: the author

Figure 18 depicts the proposed MTP-NT architecture, describing the data feed in its two branches, with the neural network highlighted with green dashed line. As seen, Branch 1 receives as input data from the neighborhood until a certain degree $N(x^{m,n})_d$, plus data from transport hubs T_{sel}. Note that to feed branch 1 with the most important features from the transportation system, data from transport hubs T are elected by a feature selection algorithm f_{sel} . The data passes through an encoding layer (Encoding step) to compression and reduce the overall size of the model and then enters the Processing Step.

Branch 2, in its turn, receives features from the base station $x^{m,n}$ (the region being analyzed at instant t, represented by $x^{m,n}(t)$ and processes at Processing Step. The branch 2 does not have a encoding or a compression layer as branch 1 to make sure no information of the main time series is lost throughout the neural network layers.

At the final step, both neighboring data and transportation features (Branch 1) and data from analyzed region (Branch 2) are pipelined through the Concatenate and output step, which performs the predictions considering both Branches 1 and 2 and concatenate.

The feature selection technique applied to the transport hubs set is represented by a condition $f_{sel}\left(x^{m,n}\left(t\right), x^{m',n'}\left(t\right)\right)$ provided by a feature selection technique that needs to be satisfied by the transport region time series. Many feature selection techniques were

tested, but the best f_{sel} is a simple picking regions under a Moore neighborhood of 20 units.

The set of BSs that covers one or more transport regions is represent as set T and the group of transport regions in T that satisfies f_{sel} is T_{sel} and can be represented as seen in Equation 11.

$$T_{sel}\left(x^{m,n}\right) = \left\{x^{m',n'} \mid f_{sel}\left(x^{m,n}(t), x^{m',n'}(t)\right) \forall x^{m',n'} \in T\right\} \tag{11}$$

As mentioned before, MTP-NT was built mostly based on LSTM, Dropout and dense layers. Dropout layers are used to avoid overfitting and each cell randomly deactivate its subsequent pair with a given rate (20% in MTP-NT training). For practical purposes, the layer resets some outputs based on the given frequency, causing some cells to not be trained randomly.

The other layers used are input layers (KERAS, 2022c), used to instantiate a Keras tensor and a concat layer (KERAS, 2022a), used to link the two branches concatenating their output. Regular densely-connected neural network layers (KERAS, 2022b) are also used in the end to output the predictions.

5.2 MTP-NT's framework architecture

The computational model can be seen in Figure 19. Here, the branch 1 receives the series from the sets $N\left(x^{m,n}\right)_d, 1 \leq d \leq 5$ and T_{sel} (when the transport data is used) over $Input\_other$ layer. Branch 2 receives the evaluated time series $x^{m,n}(t)$.

Branch 1 is composed by 5 "substeps" of layers, each one fulfilling a specific objective. First, the data passes through input layer $Input\_other$ and a dense layer $Encoding\_other\_1$, to encode the input data and begin the compression of the data.

After that there is a LSTM layer, with memory capabilities that models sequential data and a dropout layer (used to avoid overfitting), respectively $LSTM\_other\_1$ and $Dropout\_other\_1$ layers. The LSTM layer is particularly important to make temporal relations of the input data and the output predictions. 144 Units were placed in this and all other LSTM layers to ensure a maximum persistence of 24 hours of information, as 6 samples every hour (1 sample every 10 minutes), results in 144 samples every day.

Another LSTM ($LSTM\_other\_2$), dense ($Decoder\_other\_1$) and dropout ($Dropout\_other\_2$) layers were placed in sequence with the same purpose of the previous substeps. The addition of a dense layer in between LSTM and dropout layers was a result of empirical tests during the development.

The following dense ($Dense\_other\_2$) and a dropout ($Dropout\_other\_3$) layers are used to learn the overall correlation between the different time series (and also reduce overfitting) and a final dense layer $Dense\_other\_3$ is used to concatenate the first branch with the second one over the concatenate ($concatenate\_639$) layer.

Branch 2, responsible to process the time series of the region of interest, don't have a encoding step, as it process less information (just one sequence of data). Although subsequent layers in this branch have fewer units (which may indicate a lower ability to make connections and refine the predictive capacity), proportionally, this branch has a higher input data volume and connections ratio than branch 1.

Branch 2 begins with a input layer ($Input_y$), followed by two pairs of LSTM and dropout layers ($LSTM1$, $Dropout1$, $LSTM2$ and $Dropout2$). As in branch 1, these pair of layers were used to make temporal relations with the memory units of LSTM, while the dropout layers were used to avoid overfitting.

Next, a dense layer $Dense1$ and a pair of dropout and dense layers ($Dropout3$ and $Dense2$) are employed. The dense-dropout pair increases the overall performance of the neural network in tests performed and are preceded by a dense layer ($Dense2$) to connect with the first layer in the next step.

Finally, the concatenate and output step receives the outputs of the two branches. In the end, the output of the model will be the forecast of the network traffic in the region of interest for the next 10 minutes. Other time intervals were considered, but the work ended up being focused on the original data interval, with other windows (such as 1-hour forecasts, for example) being explored in minor tests, although future work could explore predictions in larger windows, so that their benefits and drawbacks can be effectively analyzed.

As seen, the model has a series of processing steps, as well as a large number of trainable parameters. This means that it also has many varibles that can be changed, such as the number of neighborhoods considered and the adoption (or not) of transport hubs, for example.

Furthermore, the nature of the problem and the variety of urban ecology in a large metropolitan center means that the choice of analysis methodology is also very important in the solution development process. Therefore, in the next chapter MTP-NT evaluation process will be discussed.

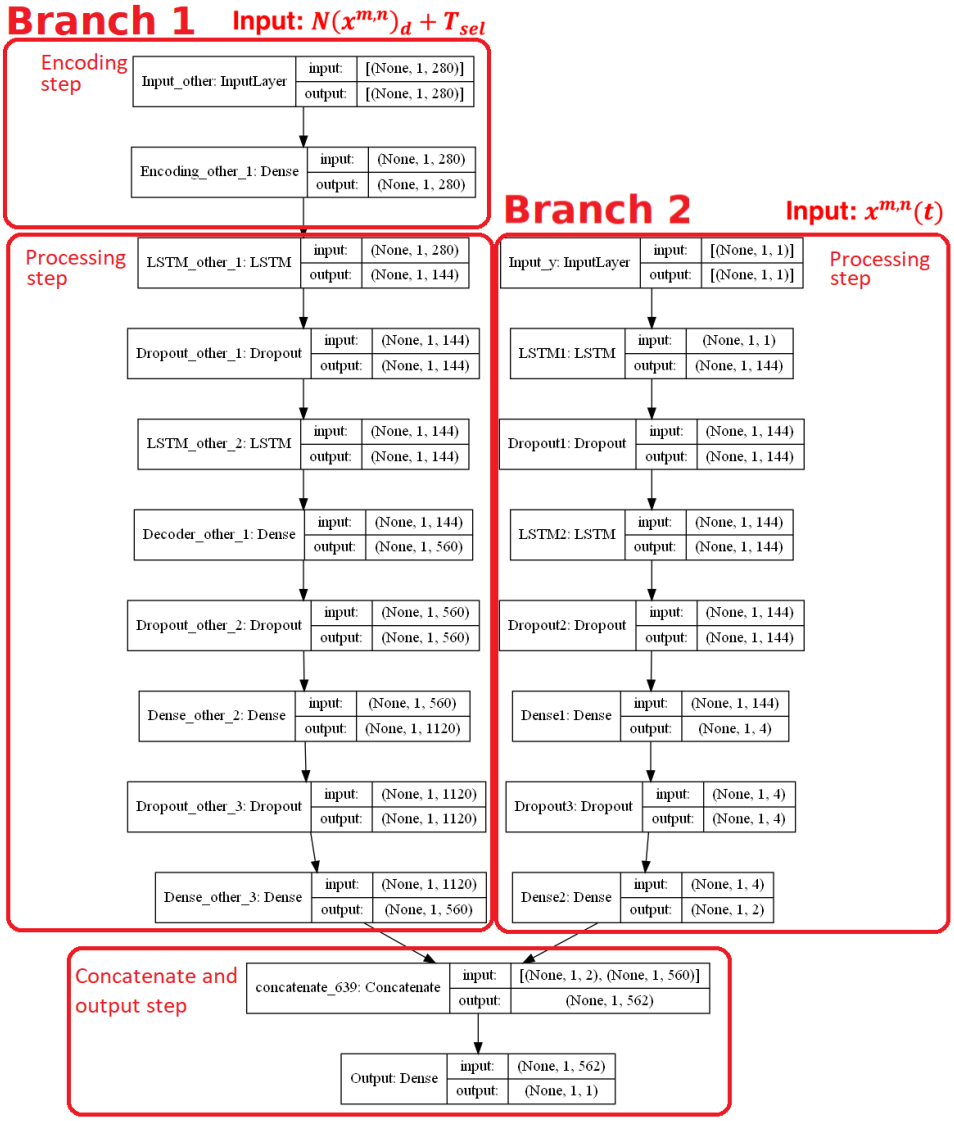


Figure 19 – Model architecture, highlighting the branches and the different stages of model processing.

Source: the author

Chapter 6

Experimental Results

MTP-NT needs to be tested in different scenarios to validate the best parameters and have the best overview of parameters. As the urban ecology can cause variations in network consumption patterns, the model have to be evaluated in a slice with diversity of the database that can reflect different network traffic patterns. To do so, 3 main group of tests will be performed: distributed tests (to capture and test the framework in different locations of the city), core tests (validating the framework performance in challenging scenarios, such as tourist centers and common areas of the city) and a event test (a mapped event with a high and aperiodic network traffic consumption pattern).

The effectiveness of public transportation data in improving prediction performance will also be tested and, therefore, these same 3 groups of tests will be replicated in models processing public transport data and others that do not handle this information.

MTP-NT is flexible, allowing variation in the volume of input data and the complexity of the model. To validate the performance of models with different magnitudes, the number of neighborhoods used in the model input (and consequently the general size of the neural networks) was varied and, to help evaluate the increase in the computational cost of these tests in relation to the improvement of predictions, the average training time of the models was used, as this metric is directly related to the computational cost of the solution.

6.1 Experimental setup

To choose the best parameters of MTP-NT, the number of neighborhoods used d was varied between 1 (considering only the immediate neighbors) and 5 (considering the neighbors from degree 1 to 5) in order to evaluate predictive performance with the neighborhood increment. Furthermore, the same executions were replicated with and without the addition of the transport hubs set T_{sel}, to evaluate the efficiency of this addition.

Some feature selection functions where tested to select the most important transport hubs. The considered functions where: F-test, Pearson correlation coefficient and a more simple test considering just the distance between the transport hubs and the region of interest, picking the 20 closest regions with transport hubs.

In order to decrease the computational cost, 64 evenly distributed regions were used for the initial tests, so that central and peripheral regions of the city in a homogeneous way would be considered. The choice of uniformly distributed points is important to contemplate city regions with different mobility dynamics and data usage patterns.

Another test done was with central and touristic areas, called "core areas", as Navigli (a district with several restaurants and touristic areas), Luigi Bocconi University, University of Milan and Duomo. This test is important to measure the effectiveness of the framework dealing with irregular and fast paced time series data, caused by the nonuniform public flow in such regions.

To confirm the effectiveness of the framework dealing with highly aperiodic data, a test was done using Giuseppe Meazza/San Siro stadium, that presents a higly aperiodic peak of network usage. This test was called "event region" and validates the good performance of MTP-NT performing a good prediction in a highly aperiodic peak of network usage in the test data, as will be presented later.

The evaluations were done using the Normalized Mean Absolute Error (NMAE), whose formula can be seen in Equation 12, where y is the target value and $\hat{y}$ is the predicted value. The NMAE was adopted because it is a proportional and self-contained metric, allowing to easily interpret a result, without needing a base for comparison.

$$NMAE(y, \hat{y}) = \frac{\sum |\hat{y} - y|}{\sum y} \tag{12}$$

The optimizer used was Adamax, which is based on Adam (Kingma; Ba, 2017). The loss function was the Mean Squared Error (MSE) and 80% of the data was used to train, with the other 20% used as test.

Finally, the execution time of the models was evaluated and the implementations were compared to performance evaluation. The scripts were executed in a machine with an Intel i5-8265U, 8GB DDR4 of RAM and a SSD SATA III with 540MB/s of reading speed.

6.2 Error evaluation

For each of the 64 regions chosen for the first test, models were built initially considering first, then first and second neighborhoods and so on, until models with all 5 degree neighbors illustrated in Figure 18. As previously mentioned, the variation in the number of neighborhoods is important to understand the tradeoff between the increment of information in the model inputs and the performance improvement.

In addition to varying the number of neighborhoods, the usage of transport hubs was also varied. Considering all combinations, a total of 640 tests were performed (64 regions with 5 different neighborhood additions and switching the transport points usage on and off).

At first, the executions were made without considering the transport hubs and varying the amount of considered neighborhoods. In Figure 20 there is an average of the results for the 64 regions. As seen, the NMAE on the executions considering just degree 1 neighbors presented a median of approximately 17.5%, while the executions considering all the 5 degree neighbors have the best performance.

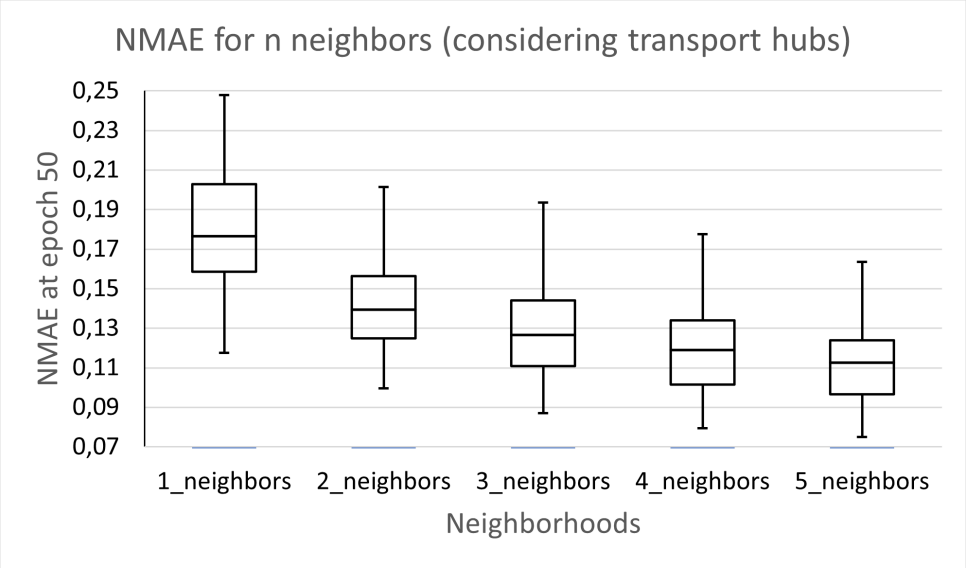


Figure 20 – Compiled from average results for the 64 regions investigated, with neighborhood data degree ranging from 1 to 5, without using data from the public transport system.

Source: the author

The results of models with transport hubs were similar to those counterparts without the transportation data with additional neighbors and, as seen in Figure 21, models with 1-4 and 1-5 moore degree neighbors showing a improvement of near 1% when compared to their peers who do not have public transport data. For executions considering just 1 degree neighbors, the upgrade tend to be better, with a mean improvement of near 9.71%, indicating that the use of transport hubs data can be an efficient way to achieve better results with less data.

In addition to the results, the processing of transport points can be important since, with the rapid development of urban transport, travel between distant regions can be done in less than 1 hour using public transport and this shows that even physically distant regions have a certain correlation in the urban mobility (and, consequently, in network usage). Besides, the observation of transport points might be important to anticipate a

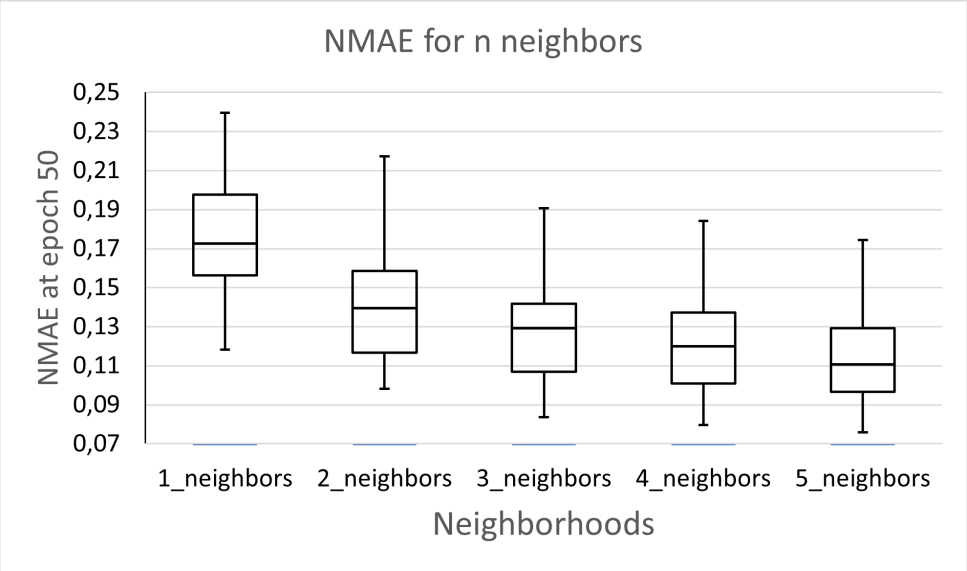


Figure 21 – Results for the 64 investigated regions, with neighbor degree varying between 1 and 5, using data from the public transport systems.

Source: the author

high traffic demand from non-seasonal events, as cited before (Wang et al., 2015). The observation of traffic in large transport hubs can be a good way to detect these unique events, although, the severe anonymization of the database used in this article prevented further investigations.

Another behavior observed was that, in peripheral points, the addition of transport regions did not imply in improvement regarding the results. On the other hand, performance tended to improve in more central regions of the city. For example, the results of the peripheral region 607 (near Vigano) without the transport system data was 11% and for the executions with the additional data was 12%. In Figure 22 the logarithmic scale Cumulative Distribution Function (CDF) of real values y and predicted values with transport hubs data $\hat{y}$ can be seen, showing that the predictions tend to be higher than the real values, reducing the chances of undersizing the network demand.

In Figure 23 the CDFs of real values and prediction of a Milan city centre can be seen in a logarithmic scale histogram. In the execution without the transport models, the NMAE was 13%, while in the execution with the transport regions, the value seen was 11%. This behavior may be explained by the feature selection adopted. As the most peripheral regions of the metropolis (rural areas, industries far from the center, for example) have fewer public transport options, the algorithm (which searches for nearby transport points) does not bring new relevant information to the model.

Region 8169 is a mall called *Centro Sarca* near *Parco Nord Milano* and there the predictions covers high peak network usage samples, while in region 607 some peak usages are not covered by any prediction. This might be related with the fact that in remote

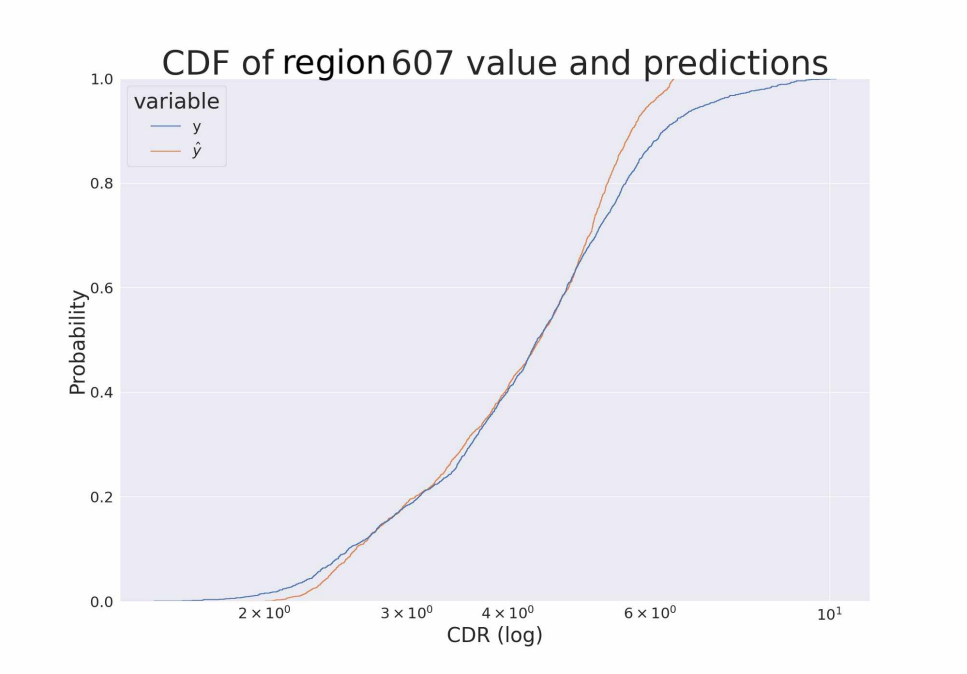


Figure 22 – Histogram of tests in region 607, showing the original CDR data and the predictions in logaritmic scale. In blue the real data and in orange the predictions with transport information.

Source: the author

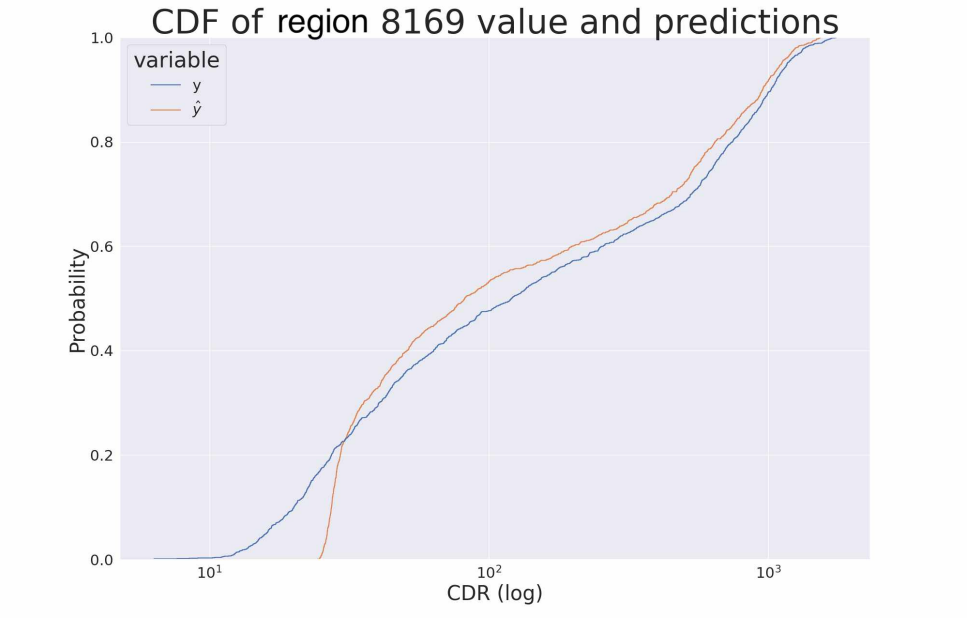


Figure 23 – Results for region 8169. In blue, the real network usage CDF and in orange the model predictions CDF.

Source: the author

regions the public transport systems are not so important/present in the mobility dynamics. Thus, in these regions, the proposed framework can attend these periods with a fixed addition to the prediction value or even a correction in the resource scheduler side to prevent demand sub-sizing and consequently drop in QoS/QoE.

Alike performance for regions with similar characteristics can be explained by the presence of patterns among different regions of the city. As seen in (Xu et al., 2017), similar regions of a city have similar network traffic patterns, related to the nature of the location (residential, commercial area, entertainment, transportation centers and comprehensive areas).

Minor tests, considering all 5 neighborhoods, where performed with 1-hour time window to validate the capabilities of MTP-NT in a different scenario than the one in which it was originally designed. As seen in Table 7, the performance in the bigger window is slightly below to 10-minute tests, even it is in the same order of magnitude.

This difference in performance may come from the reduction in the training base, since, when adjusting the data window to a range 10 times larger, there was a 10-fold decrease in the volume of the training base, compromising the model's ability to understand the context. The difference in data volume could also be observed in the model training time, which was approximately 40 seconds, while in traditional tests it was between 175 and 200 seconds (as will be explored later). Anyway, this shows that MTP-NT is sufficiently good in conditions different from those originally proposed.

Table 7 – NMAE in tests with 10-minute and 1-hour observations, varying the usage of transport hubs.

Window size	NMAE with transport data	NMAE without transport data
10 minutes	0.1120	0.1100
1 hour	0.1355	0.1441

Source: the author

6.3 Execution time evaluation

As shown in Table 6, the aggregation of more degrees of neighbors grows in the proportion of $N_x = N_{x-1} + 8(x+1)$ where x is the degree of neighborhood considered, causing a large increase in the number of regions considered. The raise in the amount of data to be processed directly implies in a higher computational cost and, consequently, in the model execution time.

The models were trained in dedicated executions, without any other process interfering in the results. The compiled algorithm execution times can be seen in Figure 24 and indicate an increase in time proportional to the increase in the number of regions used, as expected.

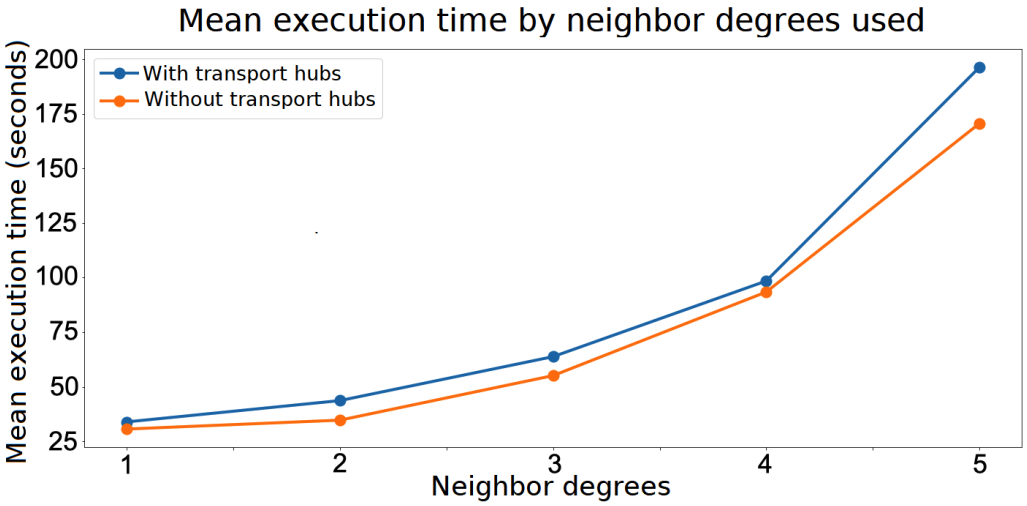


Figure 24 – Execution time for each approach.

Source: the author

However, when evaluating the same amount of neighborhoods, the addition of transport hubs did not result in a large raise in training time. This is due to the small number of series added to the model, when compared to the number of series in the neighborhoods used.

Considering the entire framework, from the information gathering from the region to the data predictions on the model itself, the increase of considered regions can imply mainly in increases on the computational resources, as more information can imply more memory demand and, in some cases, also a higher demand in processing power to maintain the same speeds for the predictions.

In general, the predictions don't demand a high execution time, taking less than 1 minute in dedicated hardware and making it viable to traffic predictions 10 minutes in advance. As the machine learning models can be accelerated over GPU executions, they can make even faster predictions in parallel execution, being possible to adapt the framework to even more complex scenarios.

Also, as MTP-NT works with smaller models making single region predictions, the predictive models can be accelerated by distributing them in VNFs and edge servers.

6.4 Benchmarking

ARIMA, Holt-Winters (HW) and LSTM are commonly used to forecast periodic time series, besides being the basis for comparison in many academic productions in the area. Prophet is also used to compare with our framework, as it is also an open source time series predictor with the focus to be easy to use (Taylor; Letham, 2018).

ARIMA is a common time series predictor used as comparison in data science studies. It is composed from autoregression (search for relationship between an observation and some lagged ones), integration (differentiation to make time series stationary) and moving average (relationship between error of previous samples and the actual sample). The tested model uses the following orders:

1. $p = 36$ using 36 lags in consideration;
2. $d = 1$ meaning one differentiation of the time series as it is stationary;
3. $q = 0$ considering no previous errors in the model.

Other configurations where tested, but these presented the best results among our tests.

HW uses additive trend and Prophet uses daily and weekly seasonality. LSTM tests use a standard scaler, that converts the inputs to a $[-1, +1]$ range and the following layers: LSTM (with 128 cells), Dropout (with 10% of dropping fraction), another LSTM with 128 cells, another dropout with 10% of fraction and a final dense layer.

Table 8 shows that the proposed model NMAE is much better than ARIMA, LSTM and Prophet at the distributed test predictions. Furthermore, an improvement is also observed when compared to HW but in a lower level. These results highlight that MTP-NT could make better predictions and work better in network traffic forecasting in any environment when compared to other prediction methods.

Table 8 – NMAE among different benchmarking techniques in Distributed, Core and Event tests.

	Distributed test	Core test	Event test
ARIMA	51.00	65.03	60.014
HW	11.78	9.34	15.16
LSTM	57.03	54.35	67.06
Prophet	61.00	94.66	178.65
MTP-NT	**11.47**	**8.22**	**11.62**

Source: the author

In (Wang et al., 2017b) the proposed framework presented a drop of 45% in error when compared to ARIMA and 62% compared to HW. The framework developed here presented a superior result in comparison with ARIMA and, nevertheless, presented a similar result seen in HW algorithm.

The main divergences in the comparisons can be attributed to characteristics such as the periodicity of the data, since HW model can only abstract periodic characteristics from a time series, not performing well in series with many non-periodic components (Chatfield, 1978). Similarly, ARIMA performs a decomposition of the periodic components of a series, which is inefficient in predicting non-periodic series (Hillmer; Tiao, 1982). It is

also important to notice that ARIMA tends to concentrate the predictions arround past series data points, making it unreliable to fast-changing scenarios (Wang et al., 2017a), what can be confirmed in the event test results.

Another point to be highlighted is that HW models do not present a forecast that precedes high frequency changes in consumption, requiring at least 1 additional sample to adjust for sudden changes, as seen in Figure 25. Here, it is possible to notice that the predictions from different HW implementations don't diverges much in a prediction from the actual values, which means that the model don't have a practical application in this scenario.

Since responding to fast changing patterns is very important to be a reliable source of network traffic predictions in the mobile network infrastructure, MTP-NT shows itself as a more reliable information platform.

The core and event tests show the same performance superiority over the conventional techniques, maintaining an NMAE close to 11%, while the other approaches showed a substantial worsening. The tests, evaluating core, central and regions with high frequency of events, proved to be important to evaluate the performance of the models at a more granular level, paying special attention to regions with atypical patterns of network consumption. Unfortunately, the related works did not map this type of events on their bases, which makes the comparison of such tests with other works harder.

The event test presented a good performance, very similar to the distributed test, as seen in Table 8. However, as seen in Figure 26, there is a completely aperiodic peak in the predicted series, demonstrating that MTP-NT would be able to uncover in advance to the infrastructure an unusual peak in network traffic. This peak, visible in the 10-minute window, would probably not be predicted in analyzes with longer windows, and could even go unnoticed and cause occasional outages in the network supply. In future works, this type of analysis can be carried out.

Similar studies don't go deeper on the general performance in different scenarios similar to what is done in the event and core tests. As shown in (Xu et al., 2017; Wang et al., 2015), however, the network consumption in different regions of the city can be highly different based on their urban ecology, presenting completely different network usage characteristics. With this, we conclude that these tests can be of paramount importance in identifying the qualities and defects of a predictive system of this nature.

The NMAE considering the transport hubs is 11.79% at epoch 50 of training, while the standard model (without transport data) results in a NMAE of 11.59% at the same epoch. Thus, this is also possible to say that, in this region, the transport hubs does not play a key role in improving model predictions.

With all the results, it can be concluded that the inclusion of public transport data, in general, improves the quality of MTP-NT predictions. When carrying out a more in-depth analysis, it is possible to conclude that central regions have a greater increase in

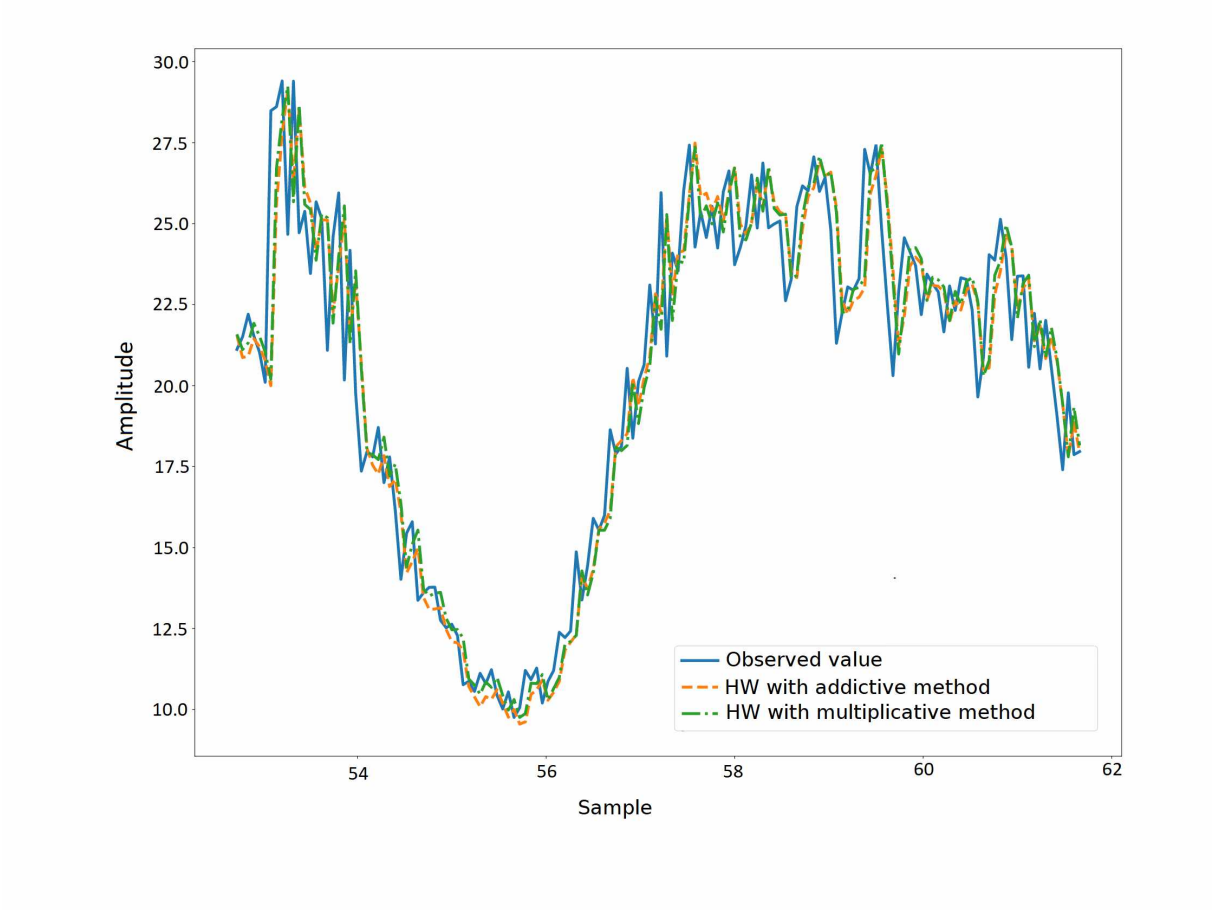


Figure 25 – Results for different HW implementations.

Source: the author

performance with this inclusion when compared to more peripheral regions of the city, as well as models considering a lower degree of neighborhood tend to have greater increases in performance.

6.5 Concluding remarks

In the 3 tests carried out (distributed, core and event tests), the results indicate that the inclusion of transport hubs tend to improve the overall performance, specially in outliers (regions with a error much above the mean). The bests results, with 5 neighbors models and transport hubs processing,presents a mean NMAE of 11%. This validates the initial theory that the inclusion of transport hubs can have a positive impact in the performance of the model, specially in highly aperiodic scenarios, as seen in the event tests. In addition to validates this theory raised in the literature, MTP-NT also proves to be important in bringing an open contribution to the study area, being an unusual practice in the segment.

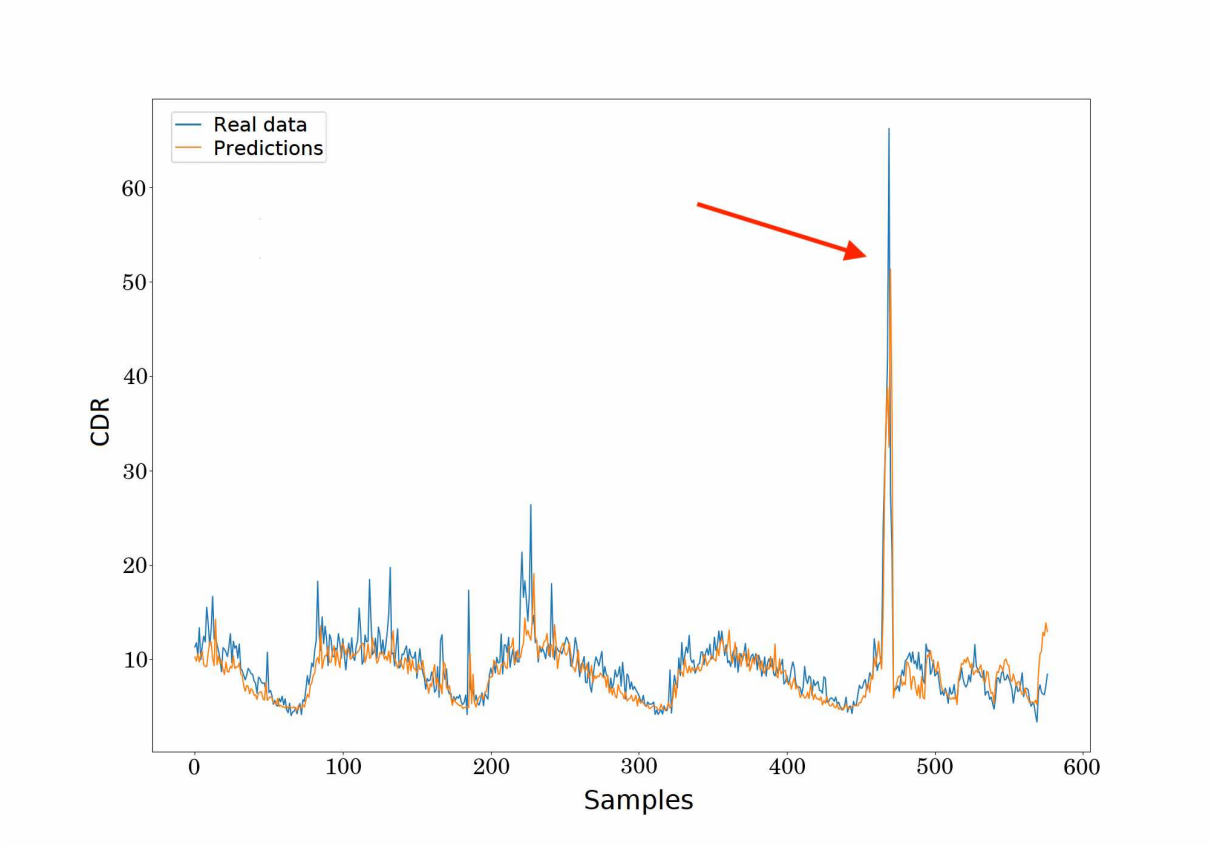


Figure 26 – Plot for San Siro/Giuseppe Meazza region predictions considering transport hubs. In blue the real network usage, in orange the model predictions with the aperiodic peak highlighted.

Source: the author

The execution time shows that, even if it comes necessary to retrain any of the models in a environment production, it can be possible to retrain and validate a model within a 10-minute window (between one prediction and other). Also, it is possible to validate how light MTP-NT can become to supply a critical, time-limited and/or computational-limited scenario.

Also, comparing the results with (Wang et al., 2017b) it is possible that MTP-NT results are at the same levels as the results seen in the literature, with the advantage of having been validated in different scenarios and with a different load of tests.

The limitations of MTP-NT usage can be found in scenarios where it is highly important to monitor all the regions/towers of a city, as tje proposed framework make individual predictions to regions and can become complex to administrate multiple models in parallel, although this parallelism may have its advantages in a distributed processing scenario when compared to competitors that make predictions in a centralized model, such as (Wang et al., 2017b; Wang et al., 2017a).

It is also important to emphasize that all tests carried out were on a sample from a

region of a single country, restricting urban ecology to a small sample of the wide variety of scenarios that can occur in different cities and countries. Therefore, it is expected that validation of the proposed framework can occur in other databases, which is feasible given that all the work is open source.

As a result, MTP-NT shows as a solid candidate for practical use in the operations of a telecommunications operator, as it was possible to validate its performance on real world data (with a very anonymized dataset), making it a viable tool even in countries with strict privacy policies.

Chapter 7

Final considerations and future work

Future mobile networking technologies may be more data driven and require robust models to support such deployments. In this way, the forecast of network demand plays a central role as it allows smarter planning and operation of the coverage regarding metropolitan regions (Sun et al., 2019).

Large cities and urban centers, in general, have a very dynamic scenario of network usage. In addition to having non-homogeneous patterns (with few regions representing a large peak in consumption), the patterns have a considerable irregular factor. In this scenario, and relying on the virtualization of resources (such as NFV achitectures) within new network technologies (such as 5G), intelligent algorithms, which anticipate network demand, can help in the resource allocation strategies performed by the scheduler. This way, both agreed QoS/QoE metrics are guaranteed while an economy of infrastructure costs can be achieved.

It is important, however, to make sure that the proposed algorithm is compatible with the network architecture proposed by 3GPP and other major network standards. To do so in 5G networks, MTP-NT was architected from the beginning with a cloud-based, VNF architecture in mind, and based in the existing architecture of NTMA. Furthermore, all data collection, storage and distribution are based on native collectors in the network, scalable databases (to handle a large volume of data) and communication sockets also native to the 5G infrastructure respectively.

Another major contribution of MTP-NT lies in the fact that all development was made available with open source licenses, as well as the database used is also publicly available. In this way, the proposal can be easily reproduced and it is expected that the adoption of this approach in this type of work will also speed up the adoption of AI in the industry, providing a standard that has already been experimentally validated.

It is also important to highlight that (Barlacchi et al., 2015) is compliant with GDPR, and by developing MTP-NT based on this standards it is ensured that privacy restrictions will not be a problem to the framework. As the major literacture explored in this dissertation relies on chinese network operators, it is possible that their implementation

in countries with rigid privacy policies will not be possible.

The main hypothesis tested in the field of framework predictions is the inclusion of public transport data, obtained through the crossing of regions and metro/tram/bus stops in the city of Milan, the focus of the work. Some academic works had already discussed the aggregation of transport data as a way of bringing a new type of information into predictive models, focusing on aperiodic events and the general improvement of forecasts, but MTP-NT is the first proposal until then to have this feature.

In the results, when placed side by side with (Wang et al., 2017a), it is observed that MTP-NT have a similar performance gain when compared to ARIMA (around 50%). Although the framework presented processes the entire array of base stations, a large amount of calculations is performed to build the autoencoders, which can present a higher computational cost.

In addition to use public transport data network traffic, which can be a promising approach to anticipate non-recurring events, MTP-NT has an interesting relationship between performance and accuracy. Thus, it can be concluded that the proposed framework presented a better performance in predicting network traffic, considering modern concepts of urban mobility and with enhancements confirmed by performance indicators.

The core and event tests showed that the proposed framework can perform well with little periodic data and even with aperiodic peaks. The event test, in particular, performed in the football stadium of Milan, validates the effectiveness of MTP-NT in extreme scenarios of network traffic by correctly predicting the large consumption spike resulting from a large-scale event within the stadium perimeter. This type of validation is particularly important, as it guarantees that the algorithm will be able to provide the scheduler with important data at times of high network demand.

Minor tests with 1-hour window shows similar performance results when compared with the original 10-minute interval. This reinforces the flexible characteristic of MTP-NT, allowing it to be adapted according to needs.

The evaluation of MTP-NT serves as a robust validation against ML limits outlined in the introduction. Firstly, in addressing the adequacy of data availability for training and evaluation purposes, our model demonstrates proficiency in handling large datasets pertinent to the problem in the same way that, as validated by the 1-hour tests, the reduction in the volume of the training base did not have major impacts on the model, making it possible to implement it in scenarios with limited data.

The utilization of pertinent information is a crucial aspect, and our approach ensures that all data incorporated into the model holds a direct causality relation to the problem by the feature selection technique. Furthermore, the response time of the final architecture has been thoroughly assessed, confirming its real-time applicability. This not only enhances the practical utility of our solution but also underscores its efficiency in addressing time-sensitive scenarios.

The monetary costs associated with GPU usage and processing is a factor that the framework addresses through its viability with different sizes, since models considering smaller number of neighborhoods are less computationally expensive (and, consequently, cheaper) than those with a larger volume. Therefore, the proposed framework is flexible enough to be implemented in different cost scenarios.

An improvement to be tested there is the increase of model size with the inclusion of new neighborhoods in the models, as the advancements in GPUs and cloud computing, which have already been reducing the computational cost to train and run increasingly large predictive models, can make larger ans faster architectures viable. In (Wang et al., 2017b), for example, neighborhoods up to 11 degrees were used. Furthermore, testing the framework on an aggregate basis based on traffic per BS instead of aggregation in fixed areas could validate the effectiveness of the model in a scenario closer to the real one.

Another proposal for the continuation of the work is the development of a technique which might forecast traffic in all parts of the city at the same time and with a minimum computational cost. A architecuture similar to (Wang et al., 2017a), that is also based on RNN combined with GSAE and LSAE, can be a way to implement this new technique.

Above all, this dissertation has a broad bibliographical review of the field of AI in mobile networks, bringing both a theoretical conceptualization of both themes as well as a general overview of the main works, showing, in general, the path that the field of network traffic forecasting is taking. However, it is important to highlight that 5G, despite being at a very advanced stage of its lifecycle, is still a field in constant evolution, just as AI, where there are new models, techniques and architectures being constantly created and improved, which means that this is a scenario of rapid change and innovation.

Bibliography

3GPP. *Architecture enhancements for 5G System (5GS) to support network data analytics services*. [S.l.], 2022. Version 17.5.0. Disponível em: <<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3579>>.

Agiwal, M.; Roy, A.; Saxena, N. Next generation 5g wireless networks: A comprehensive survey. **IEEE Communications Surveys Tutorials**, v. 18, n. 3, p. 1617–1655, 2016.

Alawe, I. et al. Improving traffic forecasting for 5g core network scalability: A machine learning approach. **IEEE Network**, v. 32, n. 6, p. 42–49, 2018.

ALLIANCE. **Green G: The Path Toward Sustainable 6G**. 2022. Accessed in 20 Dec. 2022. Disponível em: <https://www.nextgalliance.org/white_papers/green-g-the-path-towards-sustainable-6g/>.

ATM. **Mapa da rede de metrô da cidade de Milão**. 2021. Accessed in 19 Feb. 2021. Disponível em: <https://www.atm-mi.it/en/ViaggiaConNoi/Pages/SchemaReteMetro.aspx>.

Baldi, P.; Sadowski, P. The dropout learning algorithm. **Artificial Intelligence**, v. 210, p. 78–122, 2014. ISSN 0004-3702. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0004370214000216>.

Bari, F. et al. Orchestrating virtualized network functions. **IEEE Transactions on Network and Service Management**, v. 13, n. 4, p. 725–739, 2016.

Barlacchi, G. et al. A multi-source dataset of urban life in the city of milan and the province of trentino. **Sci Data 2**, 2015.

Basta, A. et al. Applying nfv and sdn to lte mobile core gateways, the functions placement problem. In: **Proceedings of the 4th workshop on All things cellular: operations, applications, & challenges**. [S.l.: s.n.], 2014. p. 33–38.

Boutaba et al. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. **Journal of Internet Services and Applications**, Springer, v. 9, n. 1, p. 1–99, 2018.

Boutaba R., S. M. L. N. e. a. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. **Journal of Internet Services and Applications 9.1**, v. 16, n. 1, p. 1–99, 2018.

Bronstein, Z.; Shraga, E. Nfv virtualisation of the home environment. In: IEEE. **2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)**. [S.l.], 2014. p. 899–904.

Chatfield, C. The holt-winters forecasting procedure. **Journal of the Royal Statistical Society: Series C (Applied Statistics)**, Wiley Online Library, v. 27, n. 3, p. 264–279, 1978.

Chen, X. et al. Analyzing and modeling spatio-temporal dependence of cellular traffic at city scale. In: **2015 IEEE International Conference on Communications (ICC)**. [S.l.: s.n.], 2015. p. 3585–3591.

Chouman, A.; Manias, D. M.; Shami, A. Towards supporting intelligence in 5g/6g core networks: Nwdaf implementation and initial analysis. **arXiv preprint arXiv:2205.15121**, 2022.

CITIVATIS. **Public Transport in Milan**. 2021. Accessed in 19 Feb. 2021. Disponível em: <https://www.introducingmilan.com/transport>.

Cybenko, G. Approximation by superpositions of a sigmoidal function. **Mathematics of control, signals and systems**, Springer, v. 2, n. 4, p. 303–314, 1989.

de Araújo, P. L.; Pasquini, R.; Murillo Guimarães Carneiro. **Milan Telecom Analysis**. 2022. Accessed in 20 Dec. 2022. Disponível em: <https://github.com/PatrickLdA/milan-telecom-analysis>.

D'Alconzo, A. et al. A survey on big data for network traffic monitoring and analysis. **IEEE Transactions on Network and Service Management**, v. 16, n. 3, p. 800–813, 2019.

ERICSSON. **Ericsson Mobility Report, Nov. 2022**. [S.l.: s.n.], 2022.

ETSI, N. **Network Function Virtualisation Use Cases**. [S.l.]: European Telecommunications Standards Institute Sophia-Antipolis, France, 2013.

Funahashi, K. On the approximate realization of continuous mappings by neural networks. **Neural networks**, Elsevier, v. 2, n. 3, p. 183–192, 1989.

Gotzner, U.; Rathgeber, R. Spatial traffic distribution in cellular networks. In: **VTC '98. 48th IEEE Vehicular Technology Conference. Pathway to Global Wireless Revolution (Cat. No.98CH36151)**. [S.l.: s.n.], 1998. v. 3, p. 1994–1998 vol.3.

Graves, A. et al. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In: **Proceedings of the International Conference on Machine Learning (ICML)**. [S.l.: s.n.], 2006. p. 369–376.

Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional lstm networks. In: **Proceedings of the International Conference on Artificial Neural Networks (ICANN)**. [S.l.: s.n.], 2005. p. 799–804.

Han, J. et al. Survey on nosql database. In: IEEE. **2011 6th international conference on pervasive computing and applications**. [S.l.], 2011. p. 363–366.

Hanyu Yang et al. A network traffic forecasting method based on sa optimized arima–bp neural network. **Computer Networks**, v. 193, p. 108102, 2021. ISSN 1389-1286. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1389128621001821>.

Herrera, J. G.; Botero, J. F. Resource allocation in nfv: A comprehensive survey. **IEEE Transactions on Network and Service Management**, IEEE, v. 13, n. 3, p. 518–532, 2016.

Hillmer, S. C.; Tiao, G. C. An arima-model-based approach to seasonal adjustment. **Journal of the American Statistical Association**, Taylor & Francis, v. 77, n. 377, p. 63–70, 1982.

Hochreiter, S.; Schmidhuber, J. Long short-term memory. **Neural computation**, v. 9, p. 1735–80, 12 1997.

Hornik, K. Approximation capabilities of multilayer feedforward networks. **Neural networks**, Elsevier, v. 4, n. 2, p. 251–257, 1991.

Hwang, J. Kk ramanrishnan, and t. wood, inetvm: High performance and flexible networning using virtualization on commodity platforms," proc. 11th usenix symp. networned syst. **Des. implement.(nsDi 14)**, v. 12, n. 1, p. 445–458, 2014.

IBM. **What are neural networks?** 2023. Accessed in 07 Sep. 2023. Disponível em: <https://www.ibm.com/topics/neural-networks>.

KERAS. **Concat layer**. 2022. Accessed in 15 Oct. 2022. Disponível em: <https://keras.io/api/layers/merging_layers/concatenate/>.

_____. **Concat layer**. 2022. Accessed in 15 Oct. 2022. Disponível em: <https://keras.io/api/layers/core_layers/dense/>.

_____. **Input object**. 2022. Accessed in 15 Oct. 2022. Disponível em: <https://keras.io/api/layers/core_layers/input/>.

KERAS. 2023. Accessed in 15 Feb. 2023. Disponível em: <https://keras.io/>.

Kim, T.; Lee, B. Scalable cdn service poc over distributed cloud management platform. In: IEEE. **2014 International Conference on Information and Communication Technology Convergence (ICTC)**. [S.l.], 2014. p. 832–833.

Kingma, D. P.; Ba, J. **Adam: A Method for Stochastic Optimization**. 2017.

Lee, D. et al. Spatial modeling of the traffic density in cellular networks. **IEEE Wireless Communications**, v. 21, n. 1, p. 80–88, 2014.

Mangili, M.; Martignon, F.; Capone, A. Stochastic planning for content delivery: Unveiling the benefits of network functions virtualization. In: IEEE. **2014 IEEE 22nd International Conference on Network Protocols**. [S.l.], 2014. p. 344–349.

Martins, J. et al. {ClickOS} and the art of network function virtualization. In: **11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)**. [S.l.: s.n.], 2014. p. 459–473.

Maulik, U.; Bandyopadhyay, S. Performance evaluation of some clustering algorithms and validity indices. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 24, n. 12, p. 1650–1654, 2002.

Niu, Z. Tango: traffic-aware network planning and green operation. **IEEE Wireless Communications**, v. 18, n. 5, p. 25–29, 2011.

OLAH, C. **Understanding LSTM Networks**. 2015. Accessed in 15 Oct. 2022. Disponível em: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>.

Pirozmand, P. et al. Human mobility in opportunistic networks: Characteristics, models and prediction methods. **Journal of Network and Computer Applications**, Elsevier, v. 42, p. 45–58, 2014.

Rankothge, W. et al. Towards making network function virtualization a cloud computing service. In: IEEE. **2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)**. [S.l.], 2015. p. 89–97.

Sciancalepore, V. et al. Mobile traffic forecasting for maximizing 5g network slicing resource utilization. In: **IEEE INFOCOM 2017 - IEEE Conference on Computer Communications**. [S.l.: s.n.], 2017. p. 1–9.

SCIKITLEARN. **Scikit-learn: machine learning in Python**. 2022. Accessed in 15 Oct. 2022. Disponível em: <https://scikit-learn.org/stable/>.

Sun, H. et al. Learning to optimize: Training deep neural networks for wireless resource management. In: **2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)**. [S.l.: s.n.], 2017. p. 1–6.

Sun, Y. et al. Application of machine learning in wireless networks: Key techniques and open issues. **IEEE Communications Surveys Tutorials**, v. 21, n. 4, p. 3072–3108, 2019.

Sutskever, I.; Vinyals, O.; Le, Q. V. Sequence to sequence learning with neural networks. In: **Advances in Neural Information Processing Systems (NIPS)**. [S.l.: s.n.], 2014. p. 3104–3112.

Taylor, S. J.; Letham, B. Forecasting at scale. **The American Statistician**, Taylor & Francis, v. 72, n. 1, p. 37–45, 2018.

Wang, H. et al. Characterizing the spatio-temporal inhomogeneity of mobile traffic in large-scale cellular data networks. In: **Proceedings of the 7th International Workshop on Hot Topics in Planet-Scale MObile Computing and Online Social NeTworking**. New York, NY, USA: Association for Computing Machinery, 2015. (HOTPOST '15), p. 19–24. ISBN 9781450335171. Disponível em: <https://doi.org/10.1145/2757513.2757518>.

Wang, J. et al. Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach. In: **IEEE INFOCOM 2017 - IEEE Conference on Computer Communications**. [S.l.: s.n.], 2017. p. 1–9.

Wang, X. et al. Spatio-temporal analysis and prediction of cellular traffic in metropolis. In: **2017 IEEE 25th International Conference on Network Protocols (ICNP)**. [S.l.: s.n.], 2017. p. 1–10.

Xu, F. et al. Understanding mobile traffic patterns of large scale cellular towers in urban environment. **IEEE/ACM Transactions on Networking**, v. 25, n. 2, p. 1147–1161, 2017.

Zen, H. et al. Statistical parametric speech synthesis using deep neural networks. In: **Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.: s.n.], 2013. p. 7962–7966.

Appendix

APPENDIX A

Readme of the source code

MTP-NT: A Mobile Traffic Predictor Enhanced by Neighboring and Transportation Data

These code are a technical analisys of A multi-source dataset of urban life in the city of Milan and the Province of Trentino paper and the development of a predictive model to forecast network traffic. The work was carried out during the master's program at the Federal University of Uberlândia.

Table of Contents

- Introduction
- Getting Started
- Technical Overview
  - Database Preprocessing
  - Libs
  - MTP-NT Compilling
  - Competitors Compilling
  - Hourly Compilling
  - Post-processing of Results
- License
  - MIT License
- Acknowledgments

Introduction

The development of techniques able to forecast the mobile network traffic in a city can feed data driven applications, as VNF orchestrators, optimizing the resource allocation and increasing the capacity of mobile networks. Despite the fact that several studies have addressed this problem, many did not consider neither

the traffic relationship among city regions nor information from public transport stations, which may provide useful information to better anticipate the network traffic.

In this work, we present a new deep learning architecture to forecast the network traffic using representation learning and recurrent neural networks. The framework, named MTP-NT, has two major components: the first responsible to learn from the time series of the region to be predicted, and the second one learning from the time series of both neighboring regions and public transportation stations. The work also reviews the 5G infrastructure based on open 3GPP specifications to explore ways to implement the framework in a real architecture. Several experiments were conducted over a dataset from the city of Milan, as well as comparisons against widely adopted and state-of-the-art techniques. The results shown in this work demonstrate that the usage of public transport information contribute to improve the forecasts in central areas of the city, as well as in regions with aperiodic demands, such as tourist regions.

Thus, this research seeks to evaluate the performance of traffic forecasting models using public data, in order to validate the performance gain with the aggregation of public transport data. The aggregation of unconventional data can be a way of adding information to the model through input that has not been explored in the scope of this research area.

The development of MTP-NT was carried out during the master's program at the Federal University of Uberlândia. The slides used in the defense, presented on 11/21/2023, can be found in the file named defesa.pdf.

Getting Started

Before execute any of the files, please install the environment listed in `requirements.txt` using pip and Anaconda.

Technical Overview

Database preprocessing

Before all model development, some pre work were done in the original database and in the collected data of public transport hubs.

misc/database_adapt.py: this code is used to take the original dataframe, that is in a format "one file per day" to a format "one region per day".

transport_modelling: contains the code to map the transport hubs in Milan. The sources used was ATM website, Wikipedia list of Milan Metro stations and Google Maps Platform. All data was compilled in transport_modelling/public_transport_locations.csv

- transport_modelling/transport_locations.py: takes a list of metro, tram and bus stations and, from the Google Maps API, saves the coordinates of the stations.
- transport_modelling/transport_locations_mapping.py: take the coordinates of every station and find the equivalent region on Milano Grid.

Libs

Some code were developed to support the models training (both MTP-NT and its competitors) in different stages. They are:

Code used in model development:

- libs/get_milano.py: a library build to get the requested data from the dataset.
- libs/functions.py: NMAE (Normalized Mean Absolute Error) and MARE (Mean absolute Relative error) implementations.

MTP-NT compilling

The MTP-NT is the purposed model, compilled by model_building.py script.

Some variables need to be attended to guarantee the work of the script:

- comms_path needs to point to repository of the data after preprocessing by misc/database_adapt.py.
- transport_path needs to point to the transport hubs data crrated by transport_modelling/transport_locations.py and transport_modelling/transport_locations_mapping.py

In lines 142--178 the region ids were the model are going to be evaluated are selected. In the end, the list of ids is stored in ids_to_use.

A print of the selected ids is saved in check_selected_ids.jpg in line 191.

transport_hubs is a list that can control the activation of transport hubs data as well as neighorrs controls wich degrees will be compilled.

After model construction and compilling, the results are saved:

- models are saved in h5 format from lines 367--371
- real values and predictions are saved in csv model from lines 381--384
- A plot of and is saved in lines 389--400
- The error csv is saved in lines 403--415

Competitors compilling

model_building_ARIMA.py: constructs ARIMA models for a selected number of regions.

model_building_HW.py: constructs Holt-Winters models for a selected number of regions.

model_building_LSTM.py: constructs LSTM models for a selected number of regions.

model_building_ARIMA.py: constructs ARIMA models for a selected number of regions.

model_building_SARIMAX.py: constructs SARIMAX models for a selected number of regions.

Hourly compilling

The original database, after compilling as described in Database Preprocessing can be recompilled again in hourly samples with the script in misc/database_adapt_hourly.py.

After all preprocessing, the resulting data also can be processed by procedures explained in MTP-NT Compilling and Competidors compilling.

Post-processing of results

Code use to validation and compilling of results:

- misc/compile_results.py: compile the results from constructed models.

License

This project is licensed under the Creative Commons 4.0.

Acknowledgments

Special thanks to the following contributors:


We would also like to express our gratitude to PPGCO-UFU for their support. And Luis Miguel Contreras Murillo as supporter of the research.