Traffic analysis and forecasting for adaptive network resource management in 5G/6G networks

Comparison of machine learning models for predicting near-future traffic demand

Oliwier Jurek
Supervisor(s): Nitinder Mohan, Marco Colocrese
EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty, Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2025

Abstract

With the exponential growth of mobile traffic in 5G networks, accurate forecasting is essential for efficient resource management. This research provides a comparative analysis of time series forecasting models for predicting near-future network traffic. Using a public dataset from a 5G base station in Barcelona, this study evaluates the performance of a traditional statistical model, against deep learning models: a Recurrent Neural Network (RNN), a Long Short-Term Memory (LSTM) and a Google timesFM model.

The results demonstrate that while the SARIMAX model struggles to capture near-future traffic demand, the deep learning approaches yield significantly higher predictive accuracy. Specifically, a simple LSTM architecture shows great results, outperforming even a more complex one. However, the timesFM model, in particular, shows the most robust generalization capabilities. Additionally, the models trained on data from one base station do not generalize well to others, highlighting significant differences in traffic characteristics even between geographically close locations. This suggests that while locally trained LSTMs are a powerful tool, future work should focus on developing more adaptive and transferable models, such as those using federated learning or graph neural networks.

1 Introduction

Since the invention and deployment of the 1G (first generation) mobile network up until today's plans of 6G the community has made significant progress. From the first generation using the analog audio signal to the recent virtualization of core architecture components, each generation has brought great improvements in capacity, functionality, and availability. The introduction of Network Function Virtualization (NFV) in 5G networks has created new opportunities for more sophisticated and at the same time more accurate traffic prediction approaches based on machine learning [1]. As noted by Chakraborty et al. "5G has a very flexible network architecture due to virtualization and will come with various customisations based on different use cases," [2]. However, this flexibility introduces significant complexity in resource management, making availability, reliability, and performance optimization increasingly challenging. The diverse requirements of different networks require intelligent traffic forecasting mechanisms that can adapt to rapidly changing network conditions and predict resource demands across multiple service types simultaneously. Recent forecasts indicate global mobile traffic will increase 10 to 100-fold between 2020 and 2030, driven primarily by video-on-demand services with high-resolution content. According to ITU-R report M.2370-0, video content alone is expected to account for two-thirds of all mobile traffic, while the global number of connected devices is projected to expand dramatically [3]. Moreover with the recent deployment and development of IoT (Internet of Things) the demand for low latency, highly reliable and flexible networks rises. 5G networks aim to achieve ambitious performance targets, such as ultra-low latency of 1ms, support for up to 1 million connected devices per square kilometer, and peak data rates reaching 20 Gbps [3]. These expectations make it more challenging to predict network traffic.

As the amount of the traffic exponentially increases, the traffic patterns have become complex. Modern mobile network traffic is characterized as a non-stationary, multivariate time series, reflecting strong temporal and spatial correlations. These features make traditional traffic engineering approaches obsolete, and accurate forecasting and dynamic management essential.

Machine learning (ML) emerges as a powerful tool to address these challenges. ML models work by using data to identify patterns, make decisions, and learn from them. Number of researchers focused on showing the efficiency of various ML models and architectures for traffic forecasting [1] [4] [5]. However, even though ML approaches were proved useful for 5G traffic prediction, a significant research gap persists regarding the practical deployment and operational efficiency of various ML models in real-world scenarios. There is a need to understand the optimal balance between a model's predictive accuracy and crucial operational factors such as its computational cost and its ability to generalize to unforeseen traffic patterns.

This research focuses on evaluating different ML models on predicting near-future traffic demand. The main contributions of this work are:

- How do different ML models compare in terms of computational resource requirements for training and deployment in 5G network traffic prediction?
- What is the optimal trade-off between model complexity and prediction accuracy for near-future traffic demand in 5G networks?
- Which model architecture demonstrates the best generalization performance on unseen traffic patterns in 5G networks?

2 Related Work

S/ARIMAX. Traditional approaches to traffic forecasting relied on statistical models like S/ARIMA/X (described in detail in Appendix A), however, these approaches are reactive, rather than anticipative, which leads to suboptimal resource management. Additionally, they require basic assumptions, such as the linearity of the time series or following a certain statistical distribution [2].

Recurrent Neural Network. (RNNs) are neural networks designed to recognize patterns in sequences of data, such as time series. Unlike feedforward neural networks, RNNs have connections that loop back. This makes them useful for tasks involving sequential data, as their output at any given step is influenced by previous inputs as shown on the figure 1. Alawe et al. showcased a forecasting approach for 5G Core Network that proved the effectiveness of recurrent neural networks in predicting the traffic demand [1].

Long Short-Term Memory. (LSTM) is an improved version of the traditional Recurrent Neural Network. It was specifically designed to address the limitations of standard RNNs, particularly the vanishing and exploding gradient problems during training.

LSTMs can retain or discard information selectively over long sequences, enabling it to capture both short-term dependencies and long-range temporal patterns more effectively. This makes LSTM

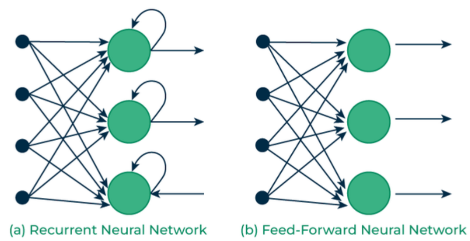

Figure 1: The difference between Recurrent Neural Networks and a Feed-Forward Neural Networks

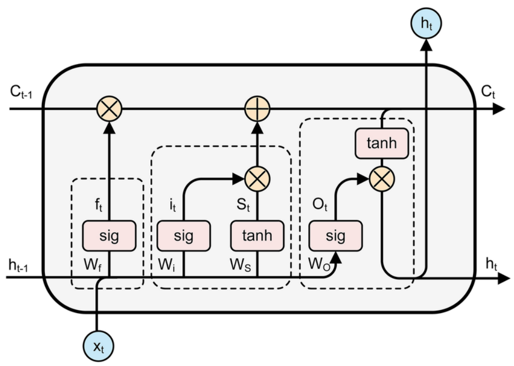

Figure 2: The architecture of LSTM cell design.

particularly suitable for modeling mobile traffic data, which often includes complex seasonal trends and bursts of activity.

This was shown to be significantly more accurate than statistical methods by Siami-Namini et al. [6] in their study. They have shown that the LSTM-based algorithm improved the prediction by 85% on average compared to ARIMA. Furthermore, the paper reports no improvement when the number of epochs is changed. Once the model has 'memorized' the dependencies in the data further training does not improve on the model's efficiency.

Another study showed that traffic forecasting LSTMs model the real-world traffic scenarios remarkably well [7]. By combining a deep neural network (DNN) with LSTM the results improved, showcasing the strength of combining neural network approaches in forecasting.

Graph Neural Networks. (GNNs) is well-suited approach to predict traffic. Jiang and Luo [8] provide a survey of GNN applications in traffic forecasting, highlighting their effectiveness in capturing the spatial-temporal dependencies inherent in network traffic. GNNs are particularly well-suited for 5G/6G network traffic prediction because they can naturally model the network topology and capture the relationships between different network elements. A great example of the above-mentioned approach is introduced Spatio-Temporal Graph Convolutional Networks (ST-GCN) by Yu et al. [4].

Federated Learning. Federated learning is used for distributed model training without sharing raw data by training model locally and then sharing the results. This is particularly relevant in 5G/6G networks, where data may be distributed across multiple base stations or network regions. Nan et al. [9] introduced a framework called region-union based federated learning for wireless traffic prediction in 5G-Advanced/6G networks. Perifanis et al. [5] conducted a comprehensive study on federated learning for 5G base station traffic forecasting, identifying key challenges including data distribution skew, quantity skew, and temporal skew. Their research showed that federated models can achieve comparable predictive accuracy to individual and centralized learning while offering improved privacy protection.

3 Methodology

3.1 Dataset

For the experiments run during this study we used the data of approximately 27,000 samples from 3 distinct base stations in Barcelona, Spain [5]. The dataset was chosen due to its diversity and abundance of data, with measurements recorded at two-minute intervals from a real LTE Physical Down-link Control Channel (PDCCH). Every sample consists of 11 features indicating the congestion of traffic, specifics of these features are detailed in the Appendix A. Our predictions in this study, however, focused on:

- **DownLink**: Represents the total volume of data transmitted from the base station to user devices during the two-minute measurement interval. This metric is indicative of the download traffic experienced by users connected to the base station.
- **UpLink**: Reversely to the DownLink traffic, UpLink corresponds to the total volume of data transmitted from user devices to the base station.

3.2 Data Preprocessing

Data preprocessing is an essential step in using machine learning and statistical models. As the predictions made based on noisy data would be inaccurate, it is crucial to perform thorough analysis and implement a data pipeline. The raw and processed data are shown on the figures 3 and 4 respectively. The data pipeline we decided to rely on was inspired by the work of Perifanis et al. [5] and is shown on the figure 5.

Outliers and Missing Data. In the dataset there are multiple outliers factoring on the model training and the succeeding evaluation. To mitigate the effects of the outliers the values below 5th quantile and above 95th have been flattened. Additionally, missing data points have been filled using forward filling technique.

Data Aggregation. The original interval of two-minutes proved too be too noisy, taking into account the need for accurate predictions for the traffic demand. We decided to run the baseline SARIMAX on the 60-minute interval and ML models on 5-minute interval. This choice will be further explained in Experiments section 5.

Scaling. Furthermore the data was scaled using MinMax function. Scaling is a common technique especially in network data, where the extremely high values influence the weights and biases of the models.

Data Split. The data from each of the base stations was split into the training set - 60%, validation set - 20% and 20% test set. This

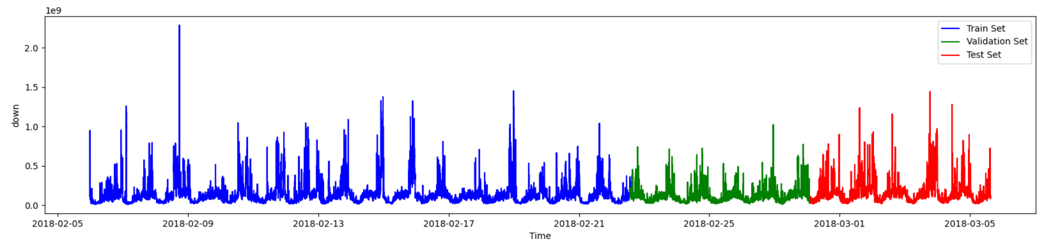

Figure 3: Original DownLink data in PobleSec base station in Barcelona, Spain.

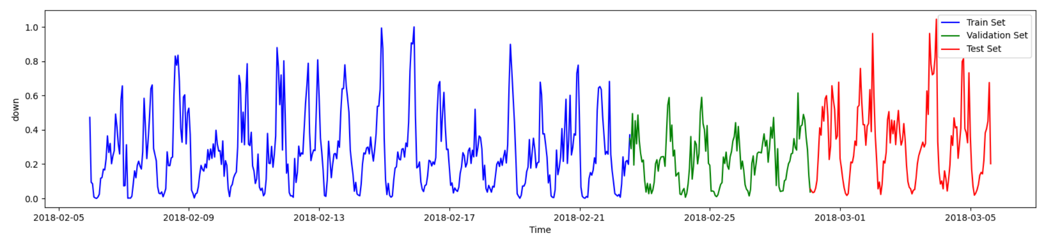

Figure 4: DownLink data after preprocessing pipeline in PobleSec base station in Barcelona, Spain.

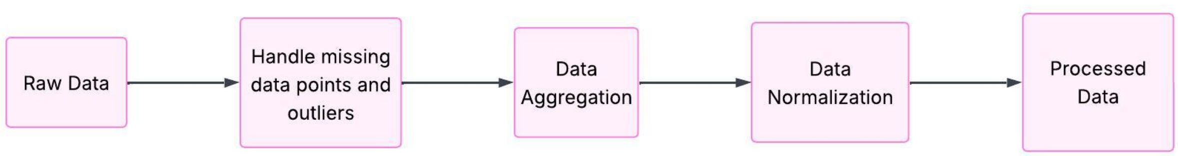

Figure 5: Data processing pipeline.

approach ensures that the models will have satisfactory results without overfitting.

Exogenous Variables. In this research it was important for the models to understand the cyclic nature of the data. By using both sine and cosine transformations on the timestamp, each time point is represented as a unique coordinate on a circle. This allows the model to understand the close relationship between time points like 11 p.m. and 1 a.m., which is critical for accurately capturing daily and weekly patterns.

3.3 Traffic Forecasting Design

The aim of the research is to compare various models in predicting near-future traffic demand. The traffic demand understood as the DownLink and UpLink features, and the near-future is the next timestep (dependent on the chosen time interval aggregation). For this several models were chosen varying by their complexity, accuracy and training time. The deep learning approaches used in this research are described in detail by Lim et al. [10].

3.4 Model Selection and Architecture

The models were selected based on the literature review and the analysis of results obtained in the literature review. Firstly the baseline models were chosen and implemented, followed with machine learning approaches.

SARIMAX. As the baseline model the statistical SARIMAX (Seasonal Auto-Regressive Integrated Moving Average with eXogenous regressors) model has been selected due to its simplicity in implementation. To make sure that the obtained results are optimal, a grid search was conducted through model parameters (both seasonal and default auto-regressive, moving average and integrated components).

RNN. For the first ML approach Recurrent Neural Network (RNN) was implemented with a structure of 2 hidden layers both of 128 nodes, which is connected to a 2 layer feed-forward neural network (256, 128 nodes). A simple model was chosen to compare the impact of model complexity by adding nodes and layers to the architecture [5].

LSTM. The selection of the LSTM was partly described in the background section 2. It is an improvement on the basic RNN cell by mitigating the exploding/vanishing gradient problem[11]. The LSTM architecture implemented in this study was adopted from the work of Nasseri et al. [12] on ambulatory seizure forecasting. This paper was chosen due to its successful application of deep LSTM networks to complex, real-world time-series data, demonstrating great forecasting performance. Both EEG signals and the detailed traffic network measurements constitute multi-variate time series data and in both domains, understanding temporal patterns and feature dependencies is essential for accurate predictions, making it a good fit for our forecasting usecase. The architecture described by Nasseri et al. [12] consists of four stacked LSTM layers. Each of these LSTM layers has 128 hidden nodes. A dropout layer with a rate of 0.2 is applied.

TimesFM. (Time Series Foundation Model) is a pretrained time-series foundation model developed by Google Research for time-series forecasting [13]. It forecasts on uni-variate time series data and due to its architecture being comprised of transformer modules it is able to adapt very well to different context lengths and diverse data. The comparison to a pretrained state-of-the-art forecasting model was useful for this study, as the computational cost for the training of this model is none.

3.5 Model Evaluation

The following metrics were used to quantify the prediction error and the overall performance of the models as proposed in [5]:

- **Mean Absolute Error (MAE)**: The MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It is the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

where n is the number of samples, y_i is the actual value, and $\hat{y}_i$ is the predicted value.

- **Root Mean Squared Error (RMSE)**: The RMSE is the square root of the MSE. It is often preferred over MSE because it has the same units as the dependent variable, making it more interpretable. It represents the standard deviation of the prediction errors.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- **Coefficient of Determination (R^2)**: The R^2 coefficient, also known as the coefficient of determination, provides a measure of how well the predicted values explain the variability of the actual values. It indicates the proportion of the variance in the dependent variable that is predictable from the independent variables. An R^2 of 1 indicates that the predictions perfectly fit the data.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where $\bar{y}$ is the mean of the actual values.

4 Experimental Setup

For this research, many experiments needed to be run simultaneously, requiring significant computational resources. To address this need, the experiments described in section 5 utilized the DelftBlue supercomputer, a resource provided by the university [14].

For GPU-accelerated experiments, such as the RNN or LSTM model training, jobs were submitted to the GPU partition. These jobs were configured to utilize one GPU per task. This setup allowed for efficient processing of computationally intensive machine learning tasks.

For CPU-bound experiments, including parameter grid searches for SARIMAX model, jobs were executed on the dedicated compute partitions. These tasks typically requested 32 CPU cores and were allocated 96 GB of RAM (3 GB per CPU), providing ample processing power for larger, parallelizable computations without requiring GPU acceleration.

5 Experiments and Training

5.1 Baseline Statistical Model

In time series forecasting with statistical models, such as SARIMAX it is crucial to determine the amount of data that the model is fit on. As the hourly aggregated data show 6, the congestion is significantly higher in the afternoon and evening than in the morning, showing strong daily patterns . Moreover, the analysis of the figure 7 suggested that there are no significant enough differences between specific days of the week to justify fitting the SARIMAX on weekly or two-weekly patterns. However, as the results on the table 1 show, there is a slight improvement from training on 1 week long dataset. Additionally, the initial experiments of fitting the model resulted in the model having problem grasping the basic patterns. Therefore we conducted an interval study to determine, which interval has the best trade-off of the length of the interval (the aim is to predict near-future traffic demand, so ideally we would keep the smallest interval) and accuracy. The table 1 is incomplete (N/A - not available) due to problems with timeout experienced on DelftBlue compute partition. The longest job runtime allowed is 24 hours, so as the training size got bigger and the interval smaller it took longer to test all the parameters for the SARIMAX model. Nonetheless, the current results show that the only viable choice is the 60-minute interval and generally results got significantly better as the data frequency interval increased. Additionally, on this interval the 1 week training/fit set performed the best by a small margin, indicating that there are some weekly patterns found by the SARIMAX model.

Training Size	Data Intervals			
	5 min	15 min	30 min	60 min
1 day	1,300,797 K	1,109,540 K	77,395 K	51,876 K
1 week	N/A	124,011 K	323,018 K	48,802 K
2 weeks	N/A	N/A	N/A	49,931 K

Table 1: Mean Absolute Error results of interval study of SARIMAX.

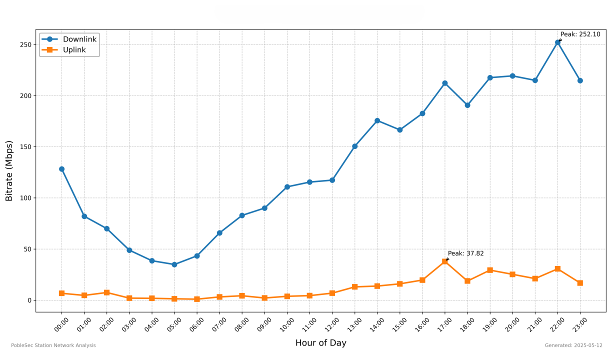

Figure 6: Aggregated hourly DownLink and UpLink bitrate for PobleSec station in Barcelona, Spain.

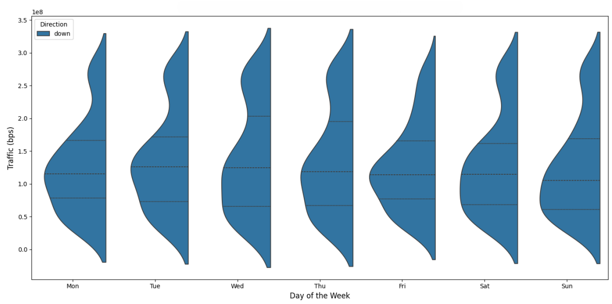

Figure 7: Weekly Distribution of Downstream Network Traffic at PobleSec Station. The width of each violin indicates the density of traffic (bps) values for that day, with dashed lines representing quartiles.

5.2 Training Machine Learning Models

The models were trained on all the features provided in the original dataset [5], additionally 3 exogenous variables were added to help the model understand the cyclical patterns of the data.

Number of Epochs. One of the most essential factors in training a well-performing model is choosing the number of epochs. The optimal number of training epochs depends on both dataset size and model complexity. Larger datasets and more complex models generally require more epochs to learn diverse patterns and converge properly, while smaller datasets and simpler models require fewer epochs. It is crucial to balance this to avoid underfitting or overfitting. For the simpler LSTM architecture the results showed that overfitting happened around epoch 100 as shown on the figure 8.

Aggregation Interval. Similariy, as with the baseline SARIMAX model it was crucial to find the optimal interval to train the model on. As the machine learning models can capture more complex patterns and handle more data than the statistical ones, the 5-minute interval performed the best, as shown on the 10.

Sequence Length. An important factor in the training of forecasting models is the sequence length, which defines how far back is the model looking to predict the next values. The models were trained

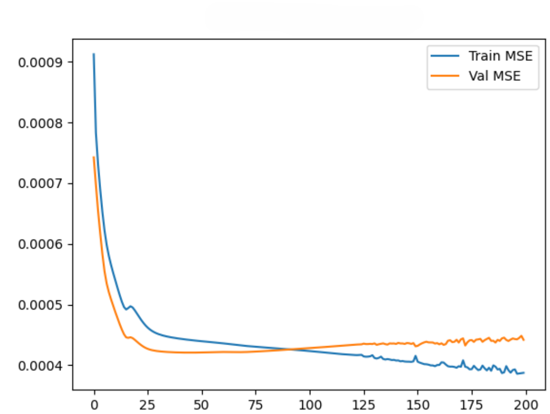

Figure 8: Train and validation loss over 200 epochs.

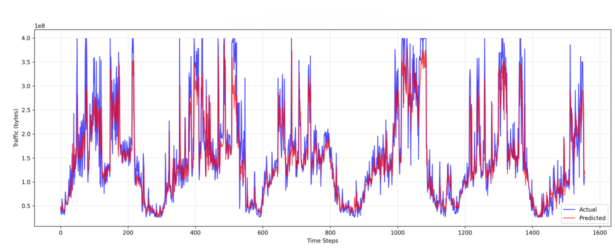

Figure 9: Predictions of LSTM model trained on 5-minute aggregated data with a 12-hour sequence length (predictions in red).

for 50 epochs using the Adam optimizer and a mean squared error loss function. The specific hyperparameters are detailed below:

- Learning Rate: 0.0001
- Dropout: 0.3
- Maximum Gradient Normalization: 5.0

They were decided upon either by inspecting the hyperparameters used in work of Perifanis et al. [5] or by testing various values independently from each other. The best performing model made its predictions based on the last 12 hours of previous data, its results are visible on the figure 9.

Sequence Length	Data Intervals			
	5 min	15 min	30 min	60 min
3 hours	32,219,820	34,141,059	37,285,673	43,748,567
6 hours	32,809,894	34,694,601	37,285,673	46,159,320
12 hours	**31,714,775**	33,825,514	36,683,636	43,748,567

Table 2: Mean Absolute Error for different sequence length in training the simple LSTM model in predicting the DownLink bitrate in PobleSec station.

Locality of Patterns. In the original dataset [5] there are data points from 3 base stations: PobleSec, LesCorts and Elborn. We conducted another series of experiments to find out how well does a model trained on one station's data perform on another's test set. To do this, we trained an LSTM on PobleSec's train set and run the forecast on LesCorts test set. For comparison we also present the MAE from a model trained on LesCorts data (model LC) and timesFM model (model TFM). As shown in the table 3, the timesFM's and LesCorts' errors are comparable, however the one trained on another's stations data performed worse. This show the effectivness of local training or using a complex pretrained model.

Model Trained On	MAE on LesCorts' Test Set
PobleSec	17,643,929
LesCorts	12,650,551
timesFM (Pre-trained)	**12,286,798**

Table 3: Comparison of Mean Absolute Error (MAE) for Three LSTM Models Evaluated on LesCorts' Test Set.

Model	Evaluation Metrics		
	MAE	RMSE	R^2
RNN - paper	34,156,115	51,659,843	0.724
LSTM - paper	33,402,828	50,769,234	0.733
LSTM - simple	**32,809,894**	**49,888,285**	**0.743**
TimesFM	33,098,485	51,145,692	0.731

Table 4: Metric comparison for different ML models for 5-minute aggregated data in predicting DownLink bitrate in PobleSec station.

Computational Cost. In terms of computational cost comparison the metric used in this study is the number of FLOPs (floating point operations) of the model. It counts every addition, subtraction, multiplication or division operation done during the runtime of the model. This metric shows the efficiency of the model before training enabling engineers to opt for a more optimal solution. The table 5 below shows as expected that the RNN model is significantly lighter than even the simple LSTM model. As the paper LSTM does not obtain better results according to the metrics in the table 4 and performs more FLOPs in total, the simple LSTM is a better fit for this particular forecasting task. Additionally, we can see that the Google TimesFM is particularly expensive in terms of FLOPs, however contrary to the paper LSTM its results prove its usefulness. TimesFM does not require training time and is adjusted to any univariate time series making it a great solution for various usecases. Moreover, timesFM can be further finetuned for a specific dataset, which improves the MAE by approximately 7%.

Analysis of Results

Machine Learning approaches for traffic demand forecasting require varying computational resources. In the section 5.2, we performed analysis based on the FLOPs (floating point operations) and parameter count. As the models get more complex, with an increasing number of parameters, so does the number of FLOPs increase, leading to a rise in the computational cost of both training and deployment.

Model	Number of FLOPs	Number of parameters
RNN - paper	40.81×10^6	76.29 K
LSTM - simple	94.33×10^6	123.46 K
LSTM - paper	359.8×10^6	470.14 K
Google TimesFM	10211.5×10^9	498,828.96 K

Table 5: Number of FLOPS and parameters for all the models

For instance, the Recurrent Neural Network (RNN) model, with 40.81×10^6 FLOPs and 76.29 K parameters, proved to be significantly "lighter" than the Long Short-Term Memory (LSTM) models. This efficiency makes RNNs a potentially more suitable option for resource-constrained environments, despite their typically lower predictive accuracy compared to more complex architectures. A notable finding from our experiments highlights that the paper inspired LSTM architecture, while possessing a higher number of FLOPs (359.8×10^6) and parameters (470.14 K), did not achieve better evaluation metrics than the "simple LSTM" model (4.33×10^6 FLOPs and 123.46 K parameters). This suggests that for the specific task of near-future traffic forecasting, simply increasing model complexity does not guarantee improved performance, and can instead lead to suboptimal resource utilization.

Conversely, the Google TimesFM model, despite its extremely high FLOPs (10211.5×10^9), offers a distinct advantage: it requires no training time as it is a pre-trained foundation model. This characteristic makes it a highly valuable solution for various use cases, particularly where rapid deployment and adaptability are crucial, outweighing its high deployment-time computational load. Understanding these trade-offs between model complexity, computational cost, and performance is essential for engineers to select the most optimal forecasting solution for 5G network traffic.

6 Responsible Research

This section outlines the ethical considerations and the steps taken to ensure the reproducibility of this research.

The primary ethical cotnsideration in network traffic analysis is user privacy. The dataset used in this study was published by Perifanis et al. [5], which consists of features related to traffic congestion. As the data is aggregated and does not have any user identification, the user privacy concerns are minimized. By using this pre-existing, public dataset, this research avoids interaction with sensitive user data. To ensure the findings of this study can be independently verified, the following methodological steps have been detailed in section 3:

- Public dataset description
- Data preprocessing pipeline
- Traffic forecasting design
- Model architecture
- Evaluation metrics used

and the specifics of the models are described in section 5:

- number of epochs
- learning rate
- dropout
- gradient normalization
- optimizer

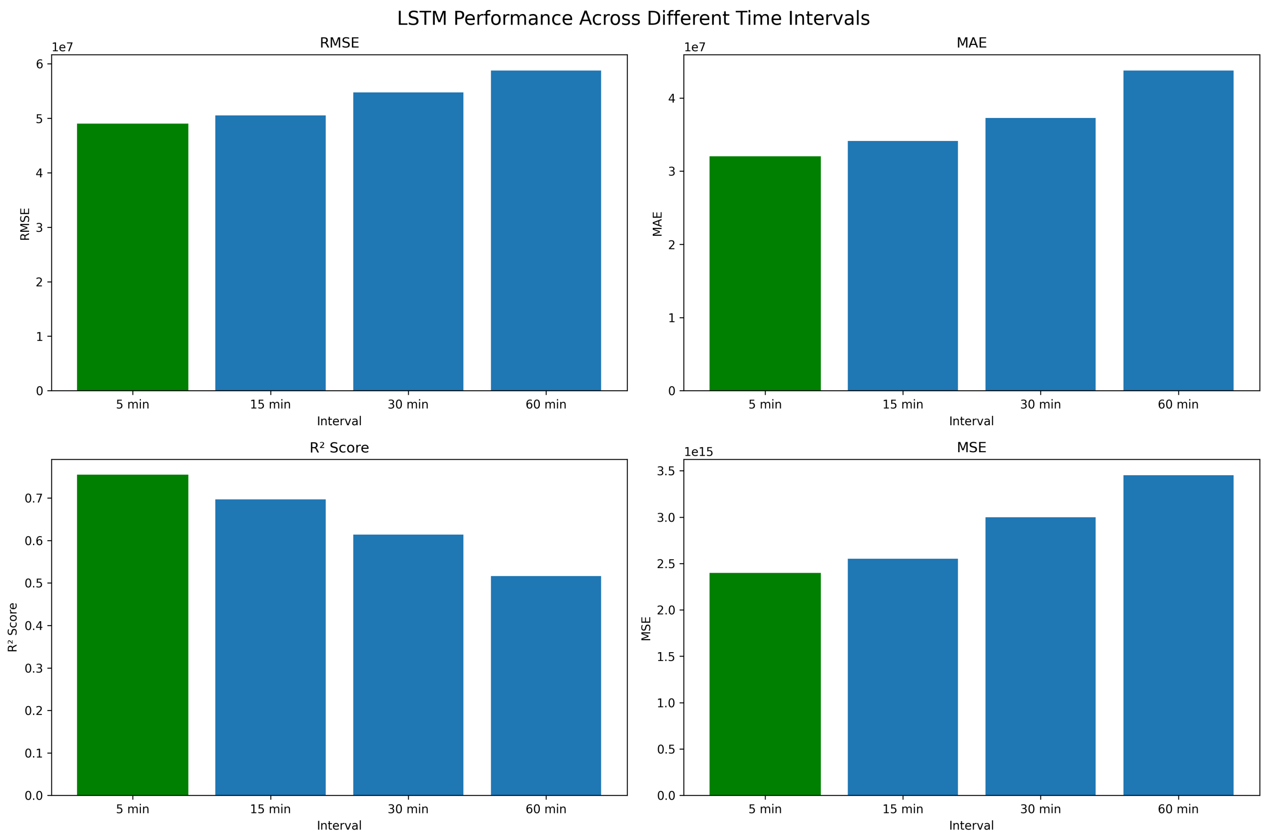

Figure 10: Evaluation metric results for different aggregation intervals for a simple 2-layer LSTM architecture.

- loss function

LLMs where used to improve flow of text and grammatical mistakes only. They didn't have any contribution to the contents of this research paper. The examplary prompts are included in the Appendix B.

7 Conclusions and Future Work

This research comprehensively compared and evaluated several time series forecasting models. The baseline Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors (SARIMAX) model exhibited considerable computational overhead during parameter fitting via grid search, and its performance was limited. While it captured fundamental hourly patterns, it consistently undervalued peak traffic periods. In contrast, the Recurrent Neural Network (RNN) model significantly outperformed the baseline, achieving a solid improvement in predictive accuracy. The timesFM foundation model demonstrated the most robust generalization capabilities on previously unseen traffic patterns, however, this enhanced performance came at the cost of requiring the highest computational resources for its deployment. Furthermore, experiments, where models trained on data from one station were tested on another, yielded unsatisfactory results. This observation strongly suggests significant variety in network traffic flow characteristics, even among stations within the same city. Future work might include adjusting the codebase for federated learning, which would enable using one global model to predict the network traffic in multiple stations. Additionally, Graph Neural Networks have been proven to be effective in traffic demand prediction. [4] and could be compared to the models compared in this work.

References

[1] Imad Alawe, Adlen Ksentini, Yassine Hadjadj-Aoul, and Philippe Bertin. Improving traffic forecasting for 5g core network scalability: A machine learning approach. *IEEE Network*, 32(6):42–49, 2018.

[2] Pousali Chakraborty, Marius Corici, and Thomas Magedanz. A comparative study for time series forecasting within software 5g networks. In *14th International Conference on Signal Processing and Communication Systems (ICSPCS)*. IEEE, 2020.

[3] Jorge Navarro-Ortiz, Pablo Romero-Diaz, Sandra Sendra, Pablo Ameigeiras, Juan J Ramos-Munoz, and Juan M Lopez-Soler. A survey on 5g usage scenarios and traffic models. *IEEE Communications Surveys & Tutorials*, 22(2):905–929, 2020.

[4] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temp oral graph convolutional networks: A deep learning framework for traffic forecasting. *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3634–3640, 2018.

[5] Vasileios Perifanis, Nicos Pavlidis, Remous-Aris Koutsiamanis, and Pavlos S Efraimidis. Federated learning for 5g base station traffic forecasting. *Computer Networks*, 235, 2023.

[6] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1394–1401, 2018.

[7] YAN Han LI Qianmu QI Yong WANG Shihao, ZHUO Qinzheng. A network traffic prediction method based on lstm. *ZTE Communications*, 17(2):19–25, 2019.

[8] Weiwei Jiang and Jiayun Luo. Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications*, 2022.

[9] Jing Nan, Manman Ai, Anjie Liu, and Xue Duan. Regional-union based federated learning for wireless traffic prediction in 5g-advanced/6g network. In *2022 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, 2022.

[10] Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194):20200209, February 2021.

[11] Takudzwa Fadziso. Overcoming the vanishing gradient problem during learning recurrent neural nets (rnn). *Asian Journal of Applied Science and Engineering*, 9(1):197–208, Dec. 2020.

[12] Mona Nasseri, Tal Pal Attia, Boney Joseph, Nicholas M. Gregg, Ewan S. Nurse, Pedro F. Viana, Gregory Worrell, Matthias Dümpelmann, Mark P. Richardson, Dean R. Freestone, and Benjamin H. Brinkmann. Ambulatory seizure forecasting with a wrist-worn device using long-short term memory deep learning. *Scientific Reports*, 11:21935, 2021.

[13] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting, 2024.

[14] Delft High Performance Computing Centre (DHPC). DelftBlue Supercomputer (Phase 2). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2, 2024.

Appendix A

SARIMAX

- **S (Seasonal):** This component accounts for seasonality in the data, meaning patterns that repeat over fixed periods (e.g., daily, weekly, or yearly cycles).
- **AR (Autoregressive):** The autoregressive part indicates that the model uses the linear combination of past observations (lagged values) to predict future values. It captures the dependence between an observation and a number of lagged observations.
- **I (Integrated):** The "integrated" aspect refers to the differencing of raw observations to make the time series stationary. Differencing involves subtracting the previous observation from the current observation. This step is necessary to remove trends or seasonality that can make a time series non-stationary and difficult to model.
- **MA (Moving Average):** The moving average part signifies that the model uses the dependency between an observation and a residual error from a moving average model applied to lagged observations. It incorporates the error terms of past predictions into the current prediction.
- **X (Exogenous):** This denotes the inclusion of external variables. These are factors that are not part of the time series itself but can influence its behavior (e.g. network-related parameters in traffic forecasting).

Dataset Features

- **DownLink**: Represents the total volume of data transmitted from the base station to user devices during the two-minute measurement interval. This metric is indicative of the download traffic experienced by users connected to the base station.
- **UpLink**: Reversely to the DownLink traffic, UpLink corresponds to the total volume of data transmitted from user devices to the base station.
- **RNTIs (RNTI Count)**: The Radio Network Temporary Identifier is a unique temporary identifier assigned to each active user equipment (UE) by the base station. The RNTI Count therefore quantifies the number of active user devices currently being served by the base station during the observed period.
- **RB Up (Resource Block Up)**: Resource Blocks (RBs) are the fundamental units of radio spectrum allocation. This feature indicates the number of resource blocks specifically allocated for uplink data transmission (from UEs to the base station) within the two-minute interval.
- **RB Down (Resource Block Down)**: Reversely to the RB Up data, RB Down are the fundamental units allocated for downlink data transmission.
- **RB Up Var (Resource Block Up Variance)**: Measures the statistical variance of the number of resource blocks allocated for uplink data transmission. High variance suggests significant fluctuations or instability in uplink resource allocation over the observed interval.
- **RB Down Var (Resource Block Down Variance)**: Measures the statistical variance of the number of resource blocks allocated for downlink data transmission.
- **MCS Up (Modulation and Coding Scheme Up)**: Modulation and Coding Schemes determine the efficiency of data transmission over the wireless channel. This feature represents the average MCS level utilized for uplink transmissions, reflecting the channel quality and efficiency for data uploaded by users.
- **MCS Down (Modulation and Coding Scheme Down)**: This indicates the average or predominant MCS level utilized for downlink transmissions, reflecting the channel quality and efficiency for data downloaded by users.
- **MCS Up Var (Modulation and Coding Scheme Up Variance)**: Measures the statistical variance of the MCS levels applied to uplink transmissions. High variance may suggest dynamic radio conditions or diverse user equipment capabilities influencing uplink transmission efficiency.
- **MCS Down Var (Modulation and Coding Scheme Down Variance)**: Measures the statistical variance of the MCS levels applied to downlink transmissions. Similar to MCS Up Var, high variance can point to changing radio conditions affecting the efficiency of downlink data delivery.

Appendix B

Use of Large Language Models

In the writing process of this research paper we used the help of LLMs. The prompts used were focused on improving the grammar and the flow of the text, not the content itself, such as: "Proofread the paper, check the grammar and spelling." or "I am writing an academic paper and this section feels out of place, please identify informal sentences.". As mentioned before the LLMs were also used to help the flow of the text, to make sure that the ideas presented are connected well.