# Graph Neural Network-based Models for Mobile Network Traffic Prediction

Duc-Thinh Ngo, Ons Aouedi, Kandaraj Piamrat, Thomas Hassan, Philippe Raipin-Parvédy

# Graph Neural Network-based Models for Mobile Network Traffic Prediction

Duc-Thinh Ngo*
duc-thinh.ngo@imt-atlantique.fr
Orange Innovation
Cesson-Sévigné, France

Ons Aouedi
ons.aouedi@inria.fr
INRIA
Nantes, France

Kandaraj Piamrat
kandaraj.piamrat@inria.fr
INRIA
Nantes, France

Thomas Hassan
thomas.hassan@orange.com
Orange Innovation
Cesson-Sévigné, France

Philippe Raipin-Parvédy
philippe.raipin@orange.com
Orange Innovation
Cesson-Sévigné, France

## ABSTRACT

From the perspective of telecommunication, next-generation networks will face several challenges of a growing number of users and services, resulting in high-traffic generation with limited network resources. Traditional predictive models, with their limited capacity to extract the intricate spatial-temporal dependencies and the multifaceted topological structures of traffic data, often cannot fully understand the dynamics of traffic flows. Recognizing the potential of Graph Neural Networks (GNNs) - which have already demonstrated efficacy in road traffic prediction - this paper delves deep into a comprehensive evaluation of sophisticated GNN-based models, namely DCRNN, ASTGCN, GWN, AGCRN, GMAN, and MTGNN. Each of these models, equipped with its unique architectural innovations, adeptly captures the underlying spatio-temporal patterns inherent to network traffic data, promising a paradigm shift in how we anticipate and manage future network traffic challenges. In particular, we focus on benchmarking these models on the network traffic prediction task using the NetMob23 Data challenge dataset.

## 1 INTRODUCTION

Network traffic prediction is an essential problem in network management. Accurate traffic prediction will provide network operators with insight into the network state and suggest appropriate actions to optimize the network. Similarly, in the context of a zero-touch network, traffic prediction can continuously inform the monitoring components about future network conditions, enabling timely decision-making. Dynamic network slicing serves as a prime application. Within the core of a B5G network where network functions are fully virtualized, network slicing allows network providers to offer their services to diverse clients while maintaining the quality of service (QoS) to meet clients' specific requirements. Moreover, dynamic network slicing offers enhanced flexibility for operators to dynamically orchestrate the slicing to address evolving client needs. Consequently, accurate traffic prediction empowers dynamic network slicing by providing valuable insights about the future client usage pattern, facilitating the preparation of slicing configurations to adapt to potential client demands.

Predicting network traffic can be formulated as a time series forecasting problem where one needs to find the best forecasting model using historical traffic data to generate the most accurate future traffic predictions. However, this approach risks losing the spatial information of the data. Considering the network traffic per evolved Node B (eNodeB), e.g., one time series per eNodeB to represent the network traffic of an eNodeB, there exists certainly a spatial correlation between these time series. For instance, geographically proximate eNodeBs can exhibit similar traffic volumes due to comparable population densities. Additionally, depending on users' mobility patterns during network usage, network traffic state can be *diffused* from one eNodeB to another. Overall, the presence of spatial correlation within the network traffic prediction raises the need to exploit another dimension of the data beyond the temporal dimension alone. Consequently, spatio-temporal traffic prediction has recently emerged to address network traffic prediction.

Graph neural networks (GNNs) play a crucial role in solving the spatio-temporal prediction problem [3, 5, 7, 9–11]. Thanks to the invariance to permutation of node orders, GNNs can extract structural information from graphs and load it within node embeddings. In the context of spatio-temporal prediction, GNNs facilitates the exploitation of the data patterns along the spatial dimension, complementing the temporal characteristics.

Similarly, road traffic prediction has also been formulated as a spatio-temporal forecasting problem. Researchers have developed various frameworks harnessing GNNs to perform multi-step ahead forecasting given historical sub-sequences of transport traffic volume (or average speed) per sensor.

In this data challenge, our objective is to benchmark several spatio-temporal models on the network traffic prediction task using the NetMob23 Data challenge dataset [8]. It is important to note that most of these models have only been benchmarked on road traffic datasets. Although the two scenarios differ, the problems in both can be framed as a spatio-temporal traffic prediction task. However, due to the distinct characteristics of the network traffic dataset, it is necessary to process this data and reframe the problem within the context of spatio-temporal network traffic forecasting. Thus, our contributions in this work are the benchmark of multiple spatio-temporal traffic prediction models, which were mostly evaluated on road traffic data, on the network traffic dataset with different evaluation protocols.

*Also with Nantes University, École Centrale Nantes, IMT Atlantique, CNRS, INRIA, LS2N, UMR 6004.

# 2 METHODOLOGY

## 2.1 Problem formulation

The traffic prediction problem is defined as the prediction of the future $T_f$ timesteps given the historical $T_h$ timesteps. We denote the multivariate time series representing the entire set of traffic data as $\boldsymbol{X}$. To be short, we denote $\boldsymbol{X}^{(t)}$ as the value of the multivariate time series at the timestep $t$ and denote $\boldsymbol{X}_i$ as the univariate time series of the $i$-th variable corresponding to the $i$-th eNodeB. Finally, $\boldsymbol{X}_i^{(t)}$ is the scalar value of the $i$-th variable at timestep $t$. We also denote the temporal length of the multivariate time series as $L$ and the number of the variables as $N$. It follows that, $\boldsymbol{X}^{(t)} \in \mathbb{R}^N$ and $\boldsymbol{X}_i \in \mathbb{R}^L$. The traffic prediction can be formulated as an optimization problem, finding a function $f$ which maps from the historical timesteps to the future timesteps:

$$\arg\min_f \mathcal{L}\left(f\left(\boldsymbol{X}^{(t-T_h:t)}\right), \boldsymbol{X}^{(t:t+T_f)}\right), \tag{1}$$

where $\mathcal{L}$ is the loss function. If we take into account the traffic structure, which is represented under a graph $\mathcal{G}$, we can redefine the problem as:

$$\arg\min_f \mathcal{L}\left(f\left(\mathcal{G}, \boldsymbol{X}^{(t-T_h:t)}\right), \boldsymbol{X}^{(t:t+T_f)}\right). \tag{2}$$

## 2.2 Data selection

Firstly, the provided data was in the form of aggregated traffic distributed across regular tiles. While this format allows organizing data in a grid, it does not reflect the real-world scenario where data is collected and aggregated per eNodeB, which is essential for forecasting traffic and performing radio resource management. To address this issue, we combine the spatio-temporal data with the eNodeB map [2] and extract the network traffic only for tiles where eNodeBs are present. Since the methodology to calculate the coverage probability for each tile was not detailed, it is difficult to reverse the calculation to obtain the per-eNodeB traffic. However, we can assume that the generated traffic in tiles attached to eNodeBs maintains the traffic patterns corresponding to those eNodeBs. During experiments, we successfully extracted locations of 1499 eNodeBs. To make the experiments faster and more lightweight, we apply random pruning to remove a portion of the eNodeBs following a Bernoulli distribution with a probability parameter of $p = 0.5$. In the end, the dataset employed in our experiments consists of traffic from 729 eNodeBs located in Paris.

Secondly, although diverse mobile application traffic data is provided, it is not necessary to predict the traffic for every one of them. Therefore, we took inspiration from the Quality of Service Class Identifier (QCI) [1] to select 5 specific applications to highlight distinct classes of QoS requirements: (1) Apple Video - conversational voice/video, (2) Fortnite - real-time gaming, (3) Netflix - buffered streaming, (4) Instagram - TCP-based and live streaming and (5) Microsoft Mail - TCP-based application. This selection allows us to capture the varied demands placed on the network infrastructure, considering factors such as peak hours, geographical locations, and different types of content consumption. For example, while Microsoft Mail experiences high demand during work hours and in the workplace, Netflix may exhibit high consumption even at night, on weekends, and in residential areas. Each application has its unique usage patterns and characteristics, which contribute to the overall diversity of the filtered dataset. By incorporating these diverse applications, our study can provide valuable insights into the challenges and opportunities associated with network traffic prediction and management across different service classes.

Finally, to evaluate the performance of multiple model architectures, we adopt a dual approach. We train one model per framework to predict the traffic for each of the five applications, allowing us to assess the generalizability of the model architectures across diverse application scenarios. This approach provides insights into the model's ability to capture different traffic patterns. Additionally, we train another model for each framework to predict the aggregated traffic resulting from the combination of the five distinct application traffic patterns. This approach allows us to benchmark the models' capability to handle complex traffic dynamics.

## 2.3 Graph construction

Similar to road traffic prediction, we have two main strategies for pre-defining the graph for spatio-temporal prediction: (1) spatial proximity graph and (2) temporal similarity graph.

Similar to the road traffic datasets [7], the edge $W_{ij}$ of the spatial proximity graph is defined and weighted based on the distance between the geolocations of nodes $i$ and $j$:

$$W_{ij}^s = \begin{cases} \exp\left(-\frac{\text{dist}(i,j)^2}{\sigma^2}\right), & \text{if } \text{dist}(i,j) < \kappa \text{ and } i \neq j. \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The negative exponential function allows the rescale of the pairwise distances of the eNodeBs into $[0, 1]$ interval while $\sigma$ is a scalar to mitigate the skewness of the distance distribution. To control the sparsity of the graph, we further prune out edges leaving only $k$-nearest neighbors for each node. Finally, we symmetrize the adjacency graph by averaging it with its transpose: $W_{\text{sym}} = (W_{\text{knn}} + W_{\text{knn}}^\top)/2$. To find these parameters, we look to reduce the sparsity of the graph as far as possible while ensuring the connexity of the graph and the symmetrical distribution of node degrees. In the experiments, we set $\sigma = 5, \kappa = 4$ $(km)$, and $k = 10$. While the extensive elaboration of these hyperparameters could be done to study their influence on the final results, it is not the focus of this work.

In the same vein, we establish the edges of the temporal similarity graph by computing the pairwise dynamic time warping (DTW) distances between the one-week traffic of each eNodeB:

$$W_{ij}^t = \begin{cases} \exp\left(-\frac{\text{dtw}(i,j)}{\gamma}\right), & \text{if } i \neq j. \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Unlike the spatial proximity graph, we directly apply the $k$-nearest neighbors approach to remove edges without $\kappa$-thresholding. Additionally, we also ensure the graph's symmetry by averaging it with its transpose. Following the same methodology as in the proximity graph construction, we set $\gamma = 10^7$ and $k = 10$.

## 2.4 Spatio-temporal models

In this challenge, we select six spatio-temporal models that have demonstrated robustness in the field of road traffic prediction.

- DCRNN [7] is a spatio-temporal model, which leverages dual directional diffusion convolution to capture spatial dependencies and incorporates it into a sequence of gated recurrent units (GRUs) to further exploit the temporal dimension.
- GWN (Graph Wavenet) [10] combines self-adaptive graph convolutional networks as spatial modules with dilated causal convolutions as temporal modules.
- AGCRN [3] integrates adaptive graph convolutional networks within a sequence of GRUs to capture both node-specific spatial and temporal correlations in traffic series.
- ASTGCN [5] is a spatio-temporal model that comprises graph convolutional network (GCN) and temporal convolution, with the introduction of spatial-temporal attention modules to distill information on both dimensions.
- GMAN [11] is a transformer framework with a multi-head attention mechanism, enabling the model to extract information from both spatial and temporal dimensions.
- MTGNN [9] is an enhanced version of GWN, which incorporates self-adaptive graph convolution with mix-hop propagation layers for spatial modules and dilated inception layers for temporal modules.

It is worth noting that among the 6 models indicated, ASTGCN, GWN, and MTGNN do not require any pre-defined graph structure since they use learnable graph adjacency matrices.

# 3 EXPERIMENTS

## 3.1 Setup

### 3.1.1 *Data processing.*

We split every single time series into several subsequences of 15 timesteps (i.e., 3h45min) using sliding windows. A spatio-temporal data sample then consists of 729 subsequences corresponding to the traffic on 729 eNodeBs within the same time interval. We then arrange sequential data samples into training, validation, and testing splits with a factor of 0.7, 0.2, and 0.1 respectively. Before feeding data to the model, we normalize it using the standard scaler and data statistics, such as mean and standard deviation, calculated from the training split and de-normalized for evaluation. In each data sample, we use 12 first timesteps as historical data to predict the next 3 timesteps, i.e., $T_h = 12$ and $T_f = 3$.

### 3.1.2 *Metrics for evaluation.*

We use three metrics (RMSE, MAPE, and MAE) as presented in the equations below. RMSE measures the squared difference between predictions and ground truth, sensitive to large errors. On the other hand, MAE considers the absolute difference regardless of scale. The MAE and RMSE consider only the difference between the prediction and the ground truth regardless of their scale. In contrast, the mean average percentage error (MAPE) quantifies errors as a percentage of the ground truth. While MAPE may initially seem more intuitive as a metric, it can be misleading, especially when dealing with datasets that have a wide range of ground truth values, encompassing both extreme and near-zero values. In such cases, a forecasting model may attempt to generate a prediction within or near the middle range of the data distribution, which can be significantly larger than the ground truth value. This can result in an enormous percentage error, leading to a disproportionately high MAPE.

**Table 1: Spatial proximity graph results.**

| | | Apple Video $\times 10^4$ or % | Fortnite $\times 10^2$ or % | Instagram $\times 10^4$ or % | Microsoft Mail $\times 10^4$ or % | Netflix $\times 10^4$ or % |
|---|---|---|---|---|---|---|
| MAE | AGCRN | 3.68 | 0.61 | 7.15 | 0.52 | 5.70 |
| | ASTGCN | 3.68 | 0.61 | 6.93 | 0.52 | 5.68 |
| | DCRNN | 3.54 | 0.59 | 6.59 | 0.49 | 5.50 |
| | GMAN | 3.57 | 0.61 | 6.88 | 0.50 | 5.56 |
| | **GWN** | 3.49 | 0.59 | **6.50** | 0.49 | **5.44** |
| | MTGNN | 3.49 | 0.59 | 6.54 | 0.49 | 5.46 |
| RMSE | AGCRN | 11.16 | 2.24 | 13.70 | 2.75 | 16.05 |
| | ASTGCN | 11.03 | 2.27 | 12.99 | 2.74 | 16.09 |
| | DCRNN | 10.86 | **2.17** | 12.61 | 2.62 | 15.85 |
| | GMAN | 11.18 | 2.29 | 12.93 | 2.75 | 16.09 |
| | GWN | 10.97 | 2.18 | **12.45** | **2.59** | 15.75 |
| | MTGNN | **10.83** | 2.19 | 12.47 | 2.60 | **15.65** |
| MAPE (%) | AGCRN | 1566.52 | 265.42 | 49.46 | 136.24 | 376.95 |
| | ASTGCN | 648.20 | 277.15 | 52.33 | **91.78** | 244.37 |
| | DCRNN | 826.57 | 272.71 | 47.00 | 172.92 | 238.97 |
| | **GMAN** | **639.09** | **236.47** | 52.85 | 148.89 | **155.80** |
| | GWN | 857.23 | 237.60 | 44.05 | 123.61 | 335.25 |
| | MTGNN | 780.38 | 253.52 | **42.94** | 111.11 | 242.85 |

In practice, employing all three metrics offers a more comprehensive assessment of forecasting models, taking into account different aspects of prediction accuracy.

- Root mean square error (RMSE)

$$\mathtt{RMSE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \sqrt{\frac{1}{|\Omega|} \sum_{i \in \Omega} (x_i - \hat{x}_i)^2} \tag{5}$$

- Mean average percentage error (MAPE)

$$\mathtt{MAPE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{|\Omega|} \sum_{i \in \Omega} \left| \frac{x_i - \hat{x}_i}{x_i} \right| \tag{6}$$

- Mean absolute error (MAE)

$$\mathtt{MAE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{|\Omega|} \sum_{i \in \Omega} |x_i - \hat{x}_i| \tag{7}$$

## 3.2 Experimental results

### 3.2.1 *Spatial proximity graph.*

We summarize in the table 1 the prediction results at a 15-minute horizon for each traffic type, modeled via the spatial proximity graph, following 3 selected evaluation metrics. It can be observed that GWN and MTGNN dominate in terms of MAE and RMSE. Meanwhile, GMAN exhibits inferior errors in terms of MAPE. We also compare visually the distribution of models' predictions and the ground truth via the box plots in figure 4. By the box plots, it can be observed that ASTGCN tends to generate small predictions, especially on the Apple Video and Microsoft Mail datasets.

### 3.2.2 *Temporal similarity graph.*

Similarly, we collect quantitative results on 5 datasets with 6 listed models and summarize them in table 2. Since ASTCN, MTGNN, and GWN do not depend on the prior graph structure, their results are identical to those in table 1. However, there are definitely changes in the final results of the other models where GWN and MTGNN still outperform others according to the MAE and RMSE metrics. In terms of MAPE, enabled by the improvement of itself and the degradation of GMAN, ASTGCN becomes the best model

**Table 2: Temporal similarity graph results.**

| | | Apple Video $\times 10^4$ or % | Fortnite $\times 10^2$ or % | Instagram $\times 10^4$ or % | Microsoft Mail $\times 10^4$ or % | Netflix $\times 10^4$ or % |
|---|---|---|---|---|---|---|
| MAE | AGCRN | 3.68 | 0.61 | 7.15 | 0.52 | 5.70 |
| | ASTGCN | 3.70 | 0.61 | 7.06 | 0.52 | 5.78 |
| | DCRNN | 3.51 | 0.59 | 6.58 | 0.49 | 5.49 |
| | GMAN | 3.57 | 0.61 | 6.79 | 0.50 | 5.51 |
| | **GWN** | 3.49 | 0.59 | **6.50** | 0.49 | **5.44** |
| | MTGNN | 3.49 | 0.59 | 6.54 | 0.49 | 5.46 |
| RMSE | AGCRN | 11.16 | 2.24 | 13.70 | 2.75 | 16.05 |
| | ASTGCN | 11.05 | 2.25 | 13.37 | 2.75 | 15.98 |
| | DCRNN | 10.93 | **2.17** | 12.48 | 2.65 | 15.82 |
| | GMAN | 11.10 | 2.28 | 12.86 | 2.72 | 15.87 |
| | GWN | 10.97 | 2.18 | **12.45** | **2.59** | 15.75 |
| | MTGNN | **10.83** | 2.19 | 12.47 | 2.60 | **15.65** |
| MAPE (%) | AGCRN | 1566.52 | 265.42 | 49.46 | 136.24 | 376.95 |
| | **ASTGCN** | **539.12** | 248.73 | **38.97** | 108.02 | **194.01** |
| | DCRNN | 732.87 | 267.54 | 45.15 | **93.92** | 287.21 |
| | GMAN | 1118.64 | 241.58 | 58.04 | 215.36 | 185.75 |
| | GWN | 857.23 | **237.60** | 44.05 | 123.61 | 335.25 |
| | MTGNN | 780.38 | 253.52 | 42.94 | 111.11 | 242.85 |

according to this metric with superior accuracy in 3 datasets; Apple Video, Instagram, and Netflix.

To compare the performance of the prior-graph-dependant models in the two setups, we estimate the performance difference in percentage: $(\mathcal{L}_{\mathcal{G}_t} - \mathcal{L}_{\mathcal{G}_s})/\mathcal{L}_{\mathcal{G}_t}$ and summarize them in table 3. Not much difference occurs in terms of MAE and RMSE metrics. However, ASTGCN and DCRNN gain large accuracy boosts when switching to the use of temporal similarity graphs on 4 over 5 datasets according to MAPE. In contrast, GMAN's accuracy degrades significantly on all 5 datasets.

#### 3.2.3 *Aggregated traffic.*

We aggregate the traffic of 5 applications to have a new dataset. The performances of each model over the different metrics and prediction horizons are summarized in the table 4. From the above analysis, in this experiment, we choose to use the temporal similarity graph for ASTGCN and DCRNN, and the proximity graph for GMAN. Overall, the quantitative results show that GWN and MTGNN, again, outperform the other models. While the former excels in predicting at the 15-minute horizon, the latter exhibits superior performance at the 30- and 45-minute horizon. Additionally, GMAN is also competitive at the 45-minute horizon. These results are relatively similar to the conclusions drawn from [9, 11] where GMAN showed its potential to predict with accuracy in the long-range thanks to the design based on pure attention mechanism.

#### 3.2.4 *Comparison to baselines.*

On the aggregated traffic data, we compare spatio-temporal prediction models against two classical time series forecasting methods (see table 4): (1) **HA** (Historical average) leverages the seasonality of the time series, and uses the weighted average of previous seasons as the prediction; (2) **VAR** (Vector autoregression) uses a system of linear equations to model the evolution of multiple time series variables over time, where each variable is regressed on its own lagged values as well as the lagged values of all other variables in the system. The results from these two models give hints about the upper bound of the errors for any advanced forecasting methods.

#### 3.2.5 *Qualitative results.*

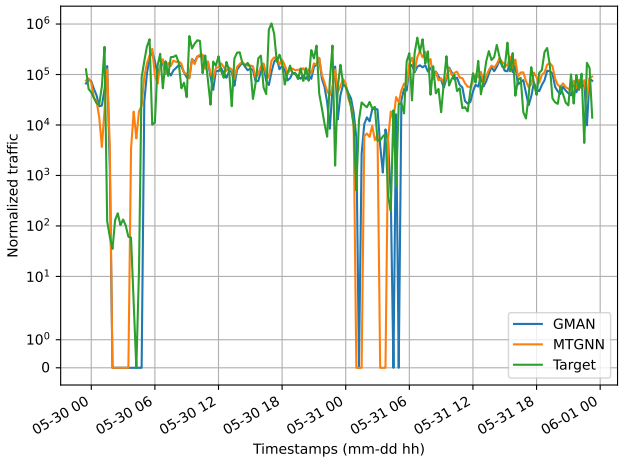



**Figure 1: A qualitative comparison between the predicted traffic by GMAN, MTGNN against the ground truth over the 2 last days of the Netflix traffic data on the tile 31649.**

According to the previous quantitative analysis, while MTGNN performs best in MAE and RMSE metrics, we achieved an even better result in MAPE with GMAN. To delve deeper into what contributes to GMAN's superior MAPE results, we turn to a qualitative examination of the predictions made by both models in comparison to the ground truth of Netflix traffic (see Fig. 1). For this qualitative analysis, we focus on the eNodeB where the difference of percentage error between the two models is the most pronounced. To ensure clarity, we visualize the traffic data for only the last 2 days. Take a look at the traffic on May 30th, where the real traffic experiences a significant drop, while both models can anticipate the drop, MTGNN "rebounds" too quickly and misses the actual traffic decline. Since the ground truth traffic is near zero and the MTGNN prediction remains at a relatively large scale (approximately $10^4$), the resulting percentage error becomes extremely large (up to $10^5$%), contributing to a MAPE much higher than GMAN. This observation holds true for various segments of traffic across different eNodeBs. In other words, GMAN excels in MAPE metric thanks to its ability to make correct predictions for near-zero traffic values.

#### 3.2.6 *Inference time.*

We benchmark the average time of each model to process a batch of a single data sample during the inference phase (Table 2). As GMAN employs a multi-head attention mechanism along both axis spatial and temporal, the computation is more complex than those using single-head attention (e.g., ASTGCN) and those leveraging pure convolution (e.g., GWN and MTGNN). Besides, DCRNN and AGCRN, owing to the incorporation of recurrent neural network (RNN) in their design, involve sequential computations that can contribute to longer inference times and greater computational demands. However, AGCRN is faster than DCRNN since it leverages the GCN, which is indeed the first-order approximation of spectral graph filtering [4, 6], while DCRNN employs the second-order approximation. Finally, ASTGCN, GWN, and MTGNN are the top 3 fastest models thanks to their lightweight design. While ASTGCN

**Table 3: Error difference (%) when switching from proximity graph to temporal similarity graph. ↓ means performance improvement and vice versa.**

| | MAE | | | MAPE | | | RMSE | | |
|---|---|---|---|---|---|---|---|---|---|
| | ASTGCN | DCRNN | GMAN | ASTGCN | DCRNN | GMAN | ASTGCN | DCRNN | GMAN |
| Apple Video | ↑ 0.64 | ↓ 0.82 | ↓ 0.19 | ↓ 16.83 | ↓ 11.34 | ↑ 75.04 | ↑ 0.15 | ↑ 0.60 | ↓ 0.77 |
| Fortnite | ↑ 0.57 | ↓ 0.11 | ↓ 0.08 | ↓ 10.26 | ↓ 1.90 | ↑ 2.16 | ↓ 0.60 | ↑ 0.13 | ↓ 0.69 |
| Instagram | ↑ 1.81 | ↓ 0.12 | ↓ 1.40 | ↓ 25.53 | ↓ 3.95 | ↑ 9.81 | ↑ 2.92 | ↓ 1.07 | ↓ 0.52 |
| Microsoft Mail | ↓ 0.42 | ↑ 0.35 | ↓ 0.02 | ↑ 17.69 | ↓ 45.68 | ↑ 44.65 | ↑ 0.16 | ↑ 1.09 | ↓ 1.17 |
| Netflix | ↑ 1.79 | ↓ 0.11 | ↓ 1.03 | ↓ 20.61 | ↑ 20.19 | ↑ 19.22 | ↓ 0.70 | ↓ 0.21 | ↓ 1.33 |

**Table 4: Multi-step prediction results on the aggregated traffic dataset.**

| | 15min | | | 30min | | | 45min | | |
|---|---|---|---|---|---|---|---|---|---|
| Metrics | MAE | MAPE | RMSE | MAE | MAPE | RMSE | MAE | MAPE | RMSE |
| HA | 18.12 | 85.62 | 37.02 | 18.12 | 85.62 | 37.02 | 18.12 | 85.62 | 37.02 |
| VAR | 17.47 | 90.09 | 33.46 | 18.19 | 97.96 | 35.25 | 18.09 | 100.09 | 35.33 |
| AGCRN | 11.76 | 45.22 | 24.66 | 13.11 | 51.61 | 27.12 | 14.07 | 58.32 | 28.83 |
| ASTGCN | 11.56 | 37.93 | 24.00 | 12.80 | 52.30 | 26.60 | 13.92 | 46.24 | 28.44 |
| DCRNN | 10.89 | 38.11 | 23.26 | 12.00 | 42.96 | 25.48 | 12.62 | 46.52 | 26.67 |
| GMAN | 11.18 | 52.33 | 23.70 | 12.04 | 55.91 | 25.29 | 12.43 | 59.13 | 26.03 |
| GWN | **10.82** | **34.79** | 23.25 | **11.86** | 38.63 | 25.32 | 12.45 | 41.66 | 26.37 |
| MTGNN | 10.96 | 34.85 | 23.25 | 11.91 | **37.45** | **25.28** | 12.43 | **40.92** | 26.33 |

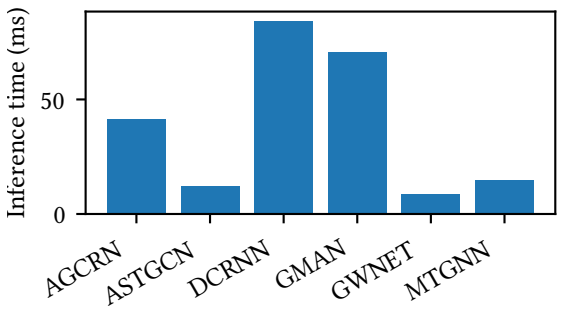



**Figure 2: Average estimated time to process one single batch of 1 data sample during inference phase.**

involves only single-head attention and GCN, GWN and MTGNN use pure convolution: temporal convolution and GCN.

## 4 DISCUSSION

Overall, on one hand, GWN and MTGNN outperform the others according to MAE and RMSE in almost every experiment. On the other hand, ASTGCN coupled with the temporal similarity graph, GMAN with the spatial proximity graph have the least mean average percentage error in predicting traffic of individual applications. Additionally, we also achieved noteworthy results with GMAN on the aggregated traffic data in the 45-minute horizon. In this regard, GMAN's performance is on par with that of MTGNN. It is also noticed that leveraging the temporal similarity graph for ASTGCN and DCRNN significantly enhances the MAPE for ASTGCN and DCRNN, leading to substantial improvements in prediction accuracy. However, applying the same strategy to GMAN has the opposite effect, causing a notable decline in prediction performance according to MAPE.

While spatio-temporal prediction is proposed to enhance the exploitation of the spatial interdependency between eNodeBs' traffic, it is not flexible by design. When a new eNodeB is installed, the eNodeB graph has to be reconstructed and the model has to be re-trained to adapt to the new setting of the dataset to perform appropriate predictions. This, however, is not a problem for road traffic prediction since the road infrastructure is more stable by nature. Therefore, a potential direction is to design new flexible spatio-temporal models that can adapt quickly to the new radio infrastructure.

## 5 CONCLUSION

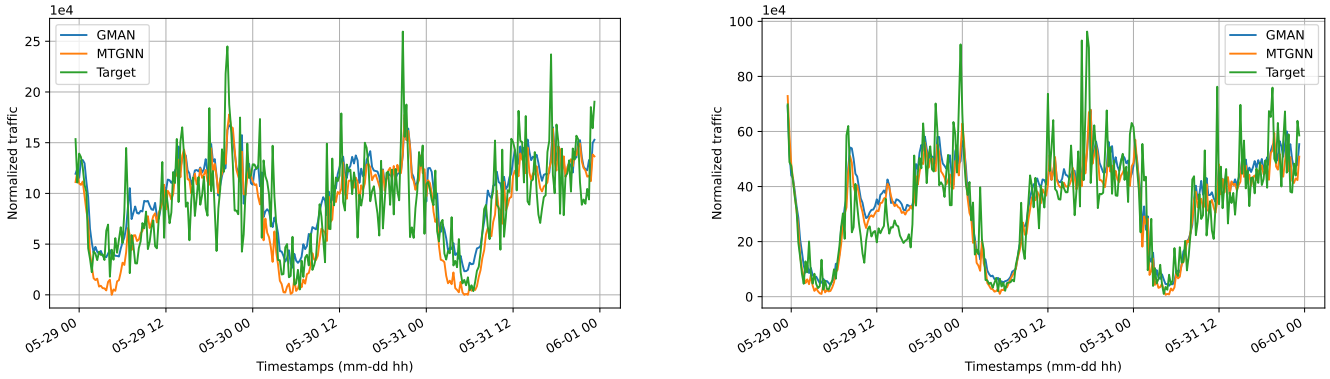
In recent years, numerous studies and efforts have been made to predict traffic. In this paper, we studied the use of several GNN-based models, a promising direction that enables intelligent network management. First, we started with a deep discussion on data selection and graph construction. Then, we provided an experimental analysis of different well-known GNN-based models. Our experiment findings reveal a contrast in the predictive efficacy of such models when applied to different traffic domains. While they showed efficient results for forecasting road traffic patterns, their performance degraded significantly when tasked with network traffic prediction. This is due to the complexity and dynamics inherent to network traffic, which may not be fully captured by models optimized for road traffic scenarios. Further investigation is essential to fine-tune and improve these models to better align with the nuances of network traffic.

## REFERENCES

[1] 2021. 3GPP TS 23.203 Policy and Charging Control Architecture V17.2.0.
[2] The National Frequency Agency. [n. d.]. Cartoradio: The map of radio sites and wave measurements. https://cartoradio.fr
[3] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems* 33 (2020), 17804–17815.
[4] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems*. https://arxiv.org/abs/1606.09375
[5] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 922–929.

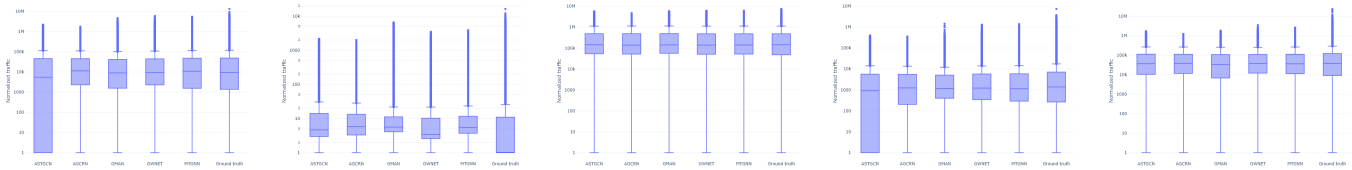
[6] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.
[7] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations (ICLR '18)*.
[8] Orlando E Martínez-Durive, Sachit Mishra, Cezary Ziemlicki, Stefania Rubrichi, Zbigniew Smoreda, and Marco Fiore. 2023. The NetMob23 dataset: A high-resolution multi-region service-level mobile data traffic cartography. arXiv:2305.06933 [cs.NI]
[9] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 753–763.
[10] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph Wavenet for Deep Spatial-Temporal Graph Modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (Macao, China) *(IJCAI'19)*. AAAI Press, 1907–1913.
[11] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 1234–1241.

# A ADDITIONAL RESULTS



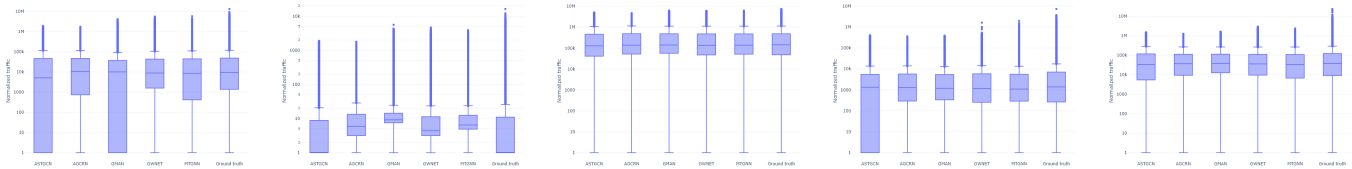

(a) Tile ID 52471 (b) Tile ID 69644

**Figure 3: MTGNN and GMAN predictions against the ground truth in the aggregated traffic dataset.**



(a) Apple-Video (b) Fortnite (c) Instagram (d) Microsoft Mail (d) Netflix

**Figure 4: Boxplots to compare predictions of all 5 models against the ground truth on each dataset when using spatial proximity graph.**



(a) Apple-Video (b) Fortnite (c) Instagram (d) Microsoft Mail (d) Netflix

**Figure 5: Boxplots to compare predictions of all 5 models against the ground truth on each dataset when using temporal similarity graph.**