This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083412, IEEE Access

Digital Object Identifier

AST-MTL: An Attention-based Multi-Task Learning Strategy for Traffic Forecasting

GIOVANNI BURONI[1], BERTRAND LEBICHOT[2], and GIANLUCA BONTEMPI.[1]
[1]Machine Learning Group, ULB, Campus de la Plaine ULB CP212, boulevard du Triomphe, 1050 Bruxelles, Belgium (e-mail:gburoni@ulb.ac.be, gbonte@ulb.ac.be)
[2]Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg

Corresponding author: Giovanni Buroni (e-mail: gburoni@ulb.ac.be).

This work is part of the project MOBIAID and was supported by Programme Opérationnel FEDER 2014-2020 de la Règion de Bruxelles Capitale and icity.brussels to strengthening Research and Innovation in Information and Communication Technologies (ICT) in Brussels.

ABSTRACT Road traffic forecasting is crucial in Intelligent Transportation Systems (ITS). To achieve accurate results, it is necessary to model the dynamic nature and the complex non-linear dependencies governing traffic. The goal is particularly challenging when the prediction involves more than just one traffic variable. This paper proposes a novel multi-task learning model, called AST-MTL, to perform multi-horizon predictions of the traffic flow and speed at the road network scale. The strategy combines a multilayer fully-connected neural network (FNN) and a multi-head attention mechanism to learn related tasks while improving generalization performance. The model also includes the graph convolutional network (GCNs) and the gated recurrent unit network (GRUs) to extract the spatial and temporal features of traffic conditions. Our experiments employ new sets of GPS data, called OBU data, to perform traffic prediction in the freeway and urban contexts. The experimental results prove our model can effectively perform multi-horizon traffic forecasting for different types of roads and outperform state-of-the-art models.

INDEX TERMS Deep learning, multi-task learning, graph mining, traffic prediction.

I. INTRODUCTION

Road transport is one of the main concerns in modern cities. This sector is responsible for different severe problems, such as pollution, road congestion, long journey times through the city, and so forth. These have negative social, environmental, and economic impacts affecting the life of citizens [1]. Nowadays, the strategy to cope with this phenomenon relies on Intelligent Transportation Systems (ITS) that integrate advanced information and communication technologies (ICT) to guarantee pro-active transportation management [1], [2].

A critical aspect of ITS is the ability to effectively predict traffic conditions of the entire road network. Generally, road traffic is expressed in terms of flow, speed, occupancy, or travel time.Systems that can accurately predict those traffic variables provide urban operators the guidance to take short and long-term actions. Over the last decades, a lot of research has been devoted to the traffic forecasting field. Over the last decades, traffic forecasting has been a vibrant field of research in both the academia and private sector. Today, predictive systems are undergoing a radical transformation thanks to the abundance of data collected in real-time by sensors deployed in urban ecosystems [1]. In particular, the appearance of Global Positioning Systems (GPS) in smartphones and vehicles has given rise to a new type of data source that gathers detailed traffic information. GPS devices send location, direction, and speed information every few seconds throughout the transportation network at low infrastructure costs, which means it is more feasible to predict traffic at a large scale.

Along with the beginning of the big data era, there have also been significant advances in the field of Artificial Intelligence (AI). At an early stage, the forecasting methods are mainly focused on modeling the temporal dynamics of traffic by adopting strict assumptions about data distribution (time-series methods) or strongly depending on handcraft feature engineering (classical machine learning methods) [1]–[4]. For these reasons, deep learning models have been gradually replacing the aforementioned. Deep Learning proved to effectively process the huge amount of mobility data and capture the non-linear spatio-temporal correlations (ST) of transportation networks without any strong assumptions [3], [5]–[7].

In this paper, we present a new spatio-temporal multi-task learning model based on attention, called AST-MTL, to perform multi-horizon network-wide traffic forecasting of traffic flow and speed. The proposed model effectively learns task

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083412, IEEE Access

shared representation through a multilayer fully-connected neural network (FNN) and a multi-head attention mechanism. At first, the FNN combines and processes multiple related tasks to extract a common representation. Then, the mechanism of attention considers together task-specific and shared representations to capture relevant information and improve the predictive performance of the model. The experimental results show the benefit of applying the attention mechanism in the context of multi-task learning. Our study is mainly inspired by the work of *Zhang et al.* [8], where the authors presented a deep learning-based multitask learning framework with Gated Recurrent Units to forecast traffic flow and traffic speed simultaneously. In particular, the paper presents the following main contributions with respect to [8]:

1) a novel multi-task learning (MTL) model that employs a multilayer fully-connected neural network (FNN) and a multi-head attention mechanism to find similarities among related tasks and improve the forecasting accuracy at road network scale. Compared to work [8], our model also accounts for the spatial component of traffic by applying stacked GCN layers.
2) an extensive experimental study based on new GPS data publicly available [1]. The model is evaluated on data sets related to both the freeway and urban road networks, which represents a novelty in the literature [1]. The results prove the competitiveness of the model compared to [8] and other counterparts. A sensitivity analysis assesses the contribution of each component to the model performance.

The rest of this paper is organized as follows. Section II reviews the current studies. Section III provides the preliminaries on multi-task learning and multi-horizon traffic forecasting on graphs. Section IV describes the methodology and introduces the AST-MTL architecture. Numerical experiments are conducted in Section V. Finally, Section VI presents the results while Section VII draws the conclusions.

II. LITERATURE REVIEW

Traffic forecasting is particularly challenging due to the temporal dynamics of traffic time-series and the complex yet unique spatial correlations of street segments.

In the past decade, a large number of methods based on deep neural networks are applied to traffic prediction problems. Most of the studies use RNNs due to their ability to memorize temporal dependencies in time-series. *Fu, Zhang, and Li* [9] compare different RNN models, namely LSTM and GRU, for traffic flow prediction. Their study proves the superiority of such models compared to ARIMA. *Cui et al.* [5] introduce a deeply stacked bidirectional and unidirectional LSTM architecture for traffic speed prediction. The model can capture both forward and backward dependencies in time-series. All these authors focus their work on the temporal component of traffic while neglecting the spatial correlations. To fill this gap, relevant is the recent works based on convolutional neural networks (CNNs) and graph neural networks (GNNs) to capture the topological dependencies in images, videos, and graphs. For example, *Cao et al.* [10] convert network-wide traffic matrices into images, after which they employ a CNN to learn the global spatial interactions and a GRU to capture the temporal features. Moreover, considering that a graph is a more appropriate abstraction of a road network, *Li et al.* [11] propose DCRNN, a new model for traffic speed prediction based on GCNs where spatial dependencies are modeled as a diffusion process. Finally, *Zhao et al.* [12] build a temporal graph convolutional network (T-GCN) model to extract the spatial and temporal dependencies simultaneously and predict the traffic volume.

In general, based on the aforementioned studies, the future seems to lie in combining CNNs/GCNs with RNNs to extract both spatial and temporal dependencies for traffic forecasting. However, one major limitation of these research works relies on the fact that the architecture of these models is designed for single-task learning (STL). This approach does not account for information shared across related tasks.

In this regard, the first attempt is carried out by *Jin et al.* [13] that introduce multi-task learning (MTL) to forecast the traffic flow. The authors design an MTL strategy based on a back-propagation network that incorporates the information of flow at several contiguous time instants. Another example is the work by *Huang et al.* [14], where they introduce a multi-task regression to predict traffic flow with a deep belief network (DBN) consisted of several neural network layers. Further, *Zhang et al.* [15] present a deep learning-based MTL model with limited neural network layers to predict network-wide traffic speed. Finally, the same authors propose [8], where a multi-task learning strategy with Gated Recurrent Units (MTL-GRU) is applied for short-term traffic flow and speed predictions. To the best of our knowledge, this last work is the only aiming at predicting two traffic variable with a MTL strategy and therefore we consider it the most closely related to ours.

In this paper, we extend previous studies by proposing AST-MTL, a novel MTL strategy based on a multilayer fully-connected neural network (FNN) and a multi-head attention mechanism to take advantage of the information shared across traffic flow and speed. Compared to previous MTL strategies, AST-MTL accounts for both the spatial and temporal components characterizing the road traffic. To validate the proposed architecture, the performance of the model is tested in the freeway and urban types of networks.

III. PRELIMINARIES

A. MULTI-TASK LEARNING

In the transportation literature, the standard methodology to tackle the problem of traffic forecasting is Single-Task Learning (STL). In STL the process of learning different traffic variables is treated as single and independents problems [5], [9], [11], [12]. This approach is limited by the fact that it does not account for the potentially rich source of information shared by related tasks. One variable could be

[1] https://www.kaggle.com/giobbu/belgium-obu

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083412, IEEE Access

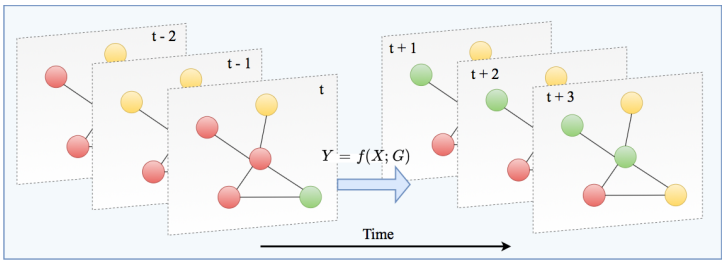


FIGURE 1: The graph-based spatial-temporal problem formulation in traffic domain. Each frame indicates the current traffic status at time step t.

an informative feature when forecasting another variable as in the case of traffic flow and speed, where a significant non-linear correlation exists [8]. Only few attempts are made to adopt Multi-task Learning (MTL) for traffic forecasting [8]. MTL is an inductive learning mechanism whose principal goal is to improve generalization performance by parallel learning different tasks through a shared representation [16]. In this work, MTL is used to learn concurrently a fixed set of M tasks [17]. In particular, we adopt the *uniform weighting strategy*, where the task-specific loss functions are added into a single function, to be minimized [18]. The loss functions of our MTL model is then:

$$L_{MTL}(\Theta) = \sum_{m=1}^{M} w^m \cdot L^m_{STL}(\Theta) \tag{1}$$

where M is the number of tasks, Θ is the set of all trainable parameters, and w^m and L^m_{STL} stands respectively for the task-specific weights (in our case w^m=1) and the loss function of the m-th task.

B. NETWORK-WIDE TRAFFIC FORECASTING TASK

The aim of traffic forecasting is to predict future traffic, given a sequence of historical traffic observations from the correlated street segments of the network. These observations are detected by sensors that monitor the traffic roads' state in real-time. We can represent the topological structure of the transportation network as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of street segments with $|\mathcal{V}| = N$ and $\mathcal{E}$ is a set of edges reflecting the connection between street segments. The connectivity information is stored in the adjacent matrix $A \in \mathbb{R}^{N\times N}$, where rows and columns are indexed by road segments, and the value of each entry indicates the connectivity: the entry value is 0 if there is no link between roads and 1 if otherwise. $X^t \in \mathbb{R}^{N\times P}$ denotes the matrix of the graph features that is observed at time t, where P is the number of features. As shown in Figure 1, the traffic prediction task can be seen as the process of learning a mapping function f from M previously observed features to H future feature matrices on the network $\mathcal{G}$:

$$Y^{t+H}_{t+1} = f(X^t_{t-(M-1)}; \mathcal{G}) \tag{2}$$

where Y^{t+H}_{t+1} denotes an array of feature matrices from time stamp i to $i+n$: $\{Y_i, Y_{i+1}, \ldots, Y_{i+n}\}$.

To fully utilize temporal information, let C^t be a set of time-based covariate vectors at time t associated to X^t. These are assumed to be known over the entire time period (e.g. day-of-the-week and hour-of-the-day). To improve the forecasting capacity, these vectors can be predetermined and wired into various locations of the model architecture:

$$Y^{t+H}_{t+1} = f([X^t_{t-(M-1)}, C^{t+H}_{t-(M-1)}]; \mathcal{G}) \tag{3}$$

IV. METHODOLOGY

In this section, we first introduce GCN and GRU used to identify the spatial and temporal relationships of traffic. Then, we described the multi-head attention mechanism proposed here as part of the multi-task learning strategy. Finally, we present the novel AST-MTL architecture to predict both traffic flow and speed.

A. SPATIAL DEPENDENCIES

It is reasonable to mathematically represent the road networks as graphs. Graph Convolutional Networks (GCNs) are a type of convolutional neural network that can work directly on graphs. In our study, we use a first-order approximation of the Laplacian graph and stack multiple localized graph convolutional layers [19]. A layer-wise linear structure is not only parameter-economic but also highly efficient for large-scale graphs, since the order of the approximation is limited to one. A multilayer GCN layer can be expressed as follow:

$$S^{(l+1)} = \sigma(\tilde{D}^{-1/2}\tilde{D}^{-1/2}S^{(l)}\Theta^{(l)}) \tag{4}$$

where $= A + Id_N$ is an adjacent matrix with self-connection structures, Id_N is an identity matrix, $\tilde{D}$ is a degree matrix, $S^{(l)} \in \mathbb{R}^{N\times l}$ is the output of layer l, $\Theta^{(l)}$ are the parameters of layer l, and $\sigma(\cdot)$ is an activation function used for nonlinear modeling.

B. TEMPORAL DEPENDENCIES

The temporal dependence is a crucial aspect for effectively forecasting the traffic state. LSTM [20] and GRU [21] are variants of RNNs that are commonly used to process mobility data and mediate the gradient vanishing and the gradient explosion problems presented by the latter [22]. In our work GRU is employed to learn the temporal variation trends of the traffic flow and speed. The model presents a more simple structure with fewer parameters than LSTM, thus

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083412, IEEE Access

resulting in faster training [23]. GRU is composed mainly of two gates: a reset gate, r, and an update gate, u. The flow of information processed by the model is introduced as follows, where S_{t-1} is the hidden state at $t-1$, X_t is the traffic speed/flow at the current moment, r determines the degree of neglecting the state information at the previous moment, while the u controls the state information quantity at the previous moment that is brought into the current state. Furthermore, c_t is the memory content stored at the current moment, and S_t is the output state at the current moment.

$$u_t = \sigma(W_u * [X_t, S_{t-1}] + b_u) \tag{5}$$

$$r_t = \sigma(W_r * [X_t, S_{t-1}] + b_r) \tag{6}$$

$$c_t = \tanh(W_c * [X_t, (r_t * S_{t-1})] + b_c) \tag{7}$$

$$S_t = u_t * S_{t-1} + (1 - u_t) * c_t \tag{8}$$

GRUs determine traffic state at the current moment by using hidden state at previous moment and traffic information at current moment as input.

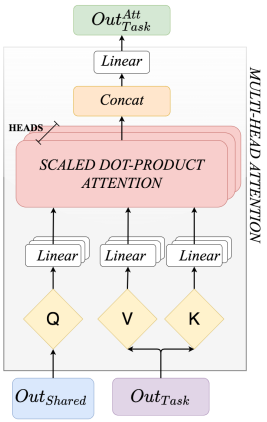


FIGURE 2: Multi-head Attention Mechanism. Several $Heads$ (attention layers) running in parallel to produce the attention-based task-specific output Out^{Att}_{Task}. Out_{Shared} represents the output of the task shared module, while Out_{Task} is the output of task-specific module.

C. MTL WITH ATTENTION

Attention mechanisms have been applied with success in various domains such as translation [24], image classification [25], and tabular learning [26]. The attention mechanism allows the models to improve the learning capacity by identifying relevant inputs portions. In the contexts of time-series prediction [27]–[29] and mobility data [29]–[31], attention has been recently applied to memorize long sequences of observations by properly incorporating the local context.

The novelty of our work stands on applying the mechanism of attention as part of the MTL strategy. Specifically, we define the attention, as defined by *Vaswani et al.* [24], to let the model learns the task-specific inputs by capturing commonalities between related tasks. A score function determines the magnitude of the attention weights based on the task-specific and shared representations. These weights serve for the model to attend only the parts of task-specific inputs that are relevant for the traffic prediction.

In general, attention mechanisms scale values V of dimension d_v based on relationships between keys K and queries Q of dimension d_{att} as follow:

$$\text{Att}(Q, K, V) = f_A(Q, K)V \tag{9}$$

where $f_A()$ is a score function. Our work employs the scaled dot-product attention [24]:

$$f_A(Q, K) = \text{Softmax}(\frac{QK^T}{\sqrt{d_{att}}}) \tag{10}$$

Notice that $f_A(Q, K)$ expresses the similarity between the content of Q and K. To improve the learning capacity, queries, keys and values can be linearly projected into O distinct subspaces [24]:

$$\text{MultiHead}(Q, K, V) = [\text{Head}_1 \cdot \text{Head}_2 \cdot ... \cdot \text{Head}_O]W_O \tag{11}$$

$$\begin{aligned} \text{Head}_o &= \text{Att}(Q_o, K_o, V_o) = \\ &\text{Att}(QW^o_Q, KW^o_K, VW^o_V) \end{aligned} \tag{12}$$

where W^o_Q, W^o_K, W^o_V, are head-specific weights matrices for keys, queries and values, while W_O linearly combines outputs concatenated $(\cdot)$ from all heads Head_o. In our study, we adopt the aforementioned mechanism to scale the values of specific tasks (V) based on the relationship between the values of the shared module (Q) and their own (K) (Figure 2).

D. PROPOSED ARCHITECTURE

The proposed AST-MTL architecture is shown in Figure 3. The model presents two consecutive levels of learning. In *Level 1*, the AST-MTL learns the specific tasks as independent problems. In *Level 2*, the model combines the tasks-specific outputs into a shared module to find a hidden common representation that improves the learning ability of each single task.

- *Level 1 - Task-Specific Learning*.
  The model learns the tasks of traffic flow and speed separately by means of spatio-temporal blocks (Figure 3a). These task-specific modules are composed of two stacked GCN layers and a GRU layer (Figure 3b). The GCN layers extract the spatial dependencies while the GRU layer the temporal ones as described in sections IV-A and IV-B.

- *Level 2 - Shared Representation Learning*.
  The model captures similarities between related tasks by means of a multilayer fully connected deep network (FNN) and temporal-attention blocks (Figure 3a) to improve the generalization of single tasks and perform multi-horizon prediction. At first, the model combines the outputs of single tasks from level 1 and passes them

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083412, IEEE Access

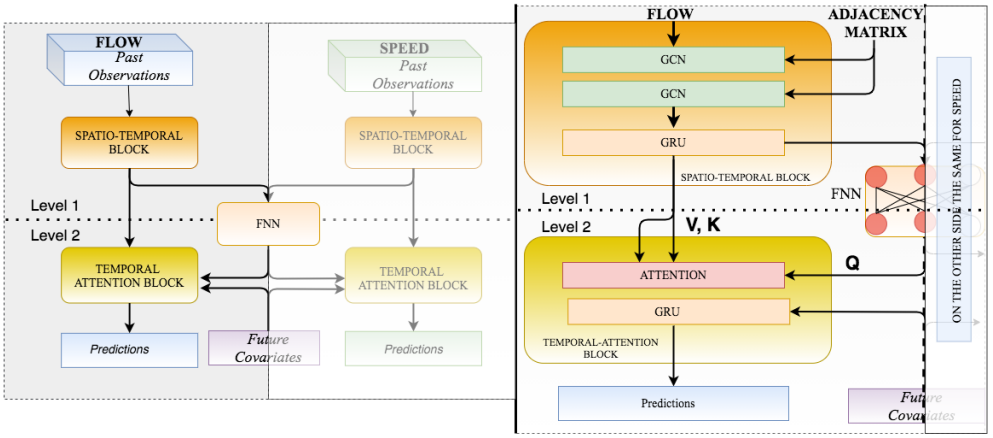


FIGURE 3: AST-MTL architecture. The framework consists of task-specific and shared modules. These are responsible for learning respectively the traffic flow (left-side of Figure (a)) and speed observations (right-side of Figure (a)). For each task, the architecture presents a spatio-temporal and a temporal-attention block to generate multiple predictions based on the past observations, the similarities between tasks, and the future time-based covariates. The FNN in the center of the architecture (Figure (a)) is the module shared across tasks.

into the FNN. The shared module takes the input information through three fully connected hidden layers to find commonalities across tasks. Accordingly, the model uses the FNN's output to improve the learning ability of task-specific representations. To pursue the goal, the model yields the information through a temporal-attention block made up of three layers (Figure 3b): an attention layer, a GRU layer, and a Dense layer. The first layer scales task-specific outputs (V) with the magnitude of the attention weights computed according to Equation 11: Q are the new values coming from FNN and the K is the single task outputs from level 1. The results are merged with future time-based covariates into a GRU layer that processes the new information for the dense layer to perform multiple predictions for the specific task.

V. EXPERIMENTS

In this section, we present the OBU data and describe the experimental setup to evaluate the performance of the AST-MTL model.

A. OBU DATA DESCRIPTION

As of 2016, all owners of Belgian lorries having a Maximum Authorized Mass exceeding 3.5 tonnes must pay a kilometer charge. Every road user who is not exempt from the toll must then install an On Board Unit (OBU) recording the distance that a lorry travels within Belgium. On average every working day, more than 150,000 trucks are detected inside the country. Each truck device sends a message approximately every 30 seconds (from 3 a.m. to 2.59 a.m. of the following day). Each OBU record contains an anonymous *Identifier* (ID resetting every day at 3 a.m.), the *Timestamp*, the *GPS Position* (latitude, longitude), the *Speed* (engine) and the *Direction* (compass). The large volume and the streaming nature of the OBU data required the setup of a big data platform for an efficient collection, storage, and analysis [32], [33].

1) Data Processing

The OBU data require a pre-processing step to predict the traffic conditions at network scale since the trucks recorded positions may refer not only to streets but also to areas where trucks perform the daily activity (e.g. loading/unloading goods, stopping at depots' parking slot).
In this regard, we first retrieved all the necessary streets from `OpenStreetMap` with `Module osmnx` [34]. Then, we map the GPS points on their corresponding road segments to obtain only the data belonging to the streets of interest. We carried out such operations for both the Belgian Freeway System and the road network of the Bruxelles-Capital Region (Figure 4). This allows testing our methodology in two different contexts [1]: the freeway and urban environment. The latter, in particular, has been defined in traffic forecasting literature as the less addressed and much more challenging [1], [2]. Table 1 summarised the main characteristics of these two types of road networks.

2) Future Covariates

As explained in Section III-B, time-based covariates allow taking full advantage of the temporal information. Such features are deterministic and therefore known in advance for future traffic predictions. Starting from $Timestamp$ variable, the following covariates are obtained:

1) $Sine$ and $Cosine$ transformation for Hours of the Day to take into account the cyclic nature of time (e.g. 0 and 23 hours are close to each other);

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083412, IEEE Access

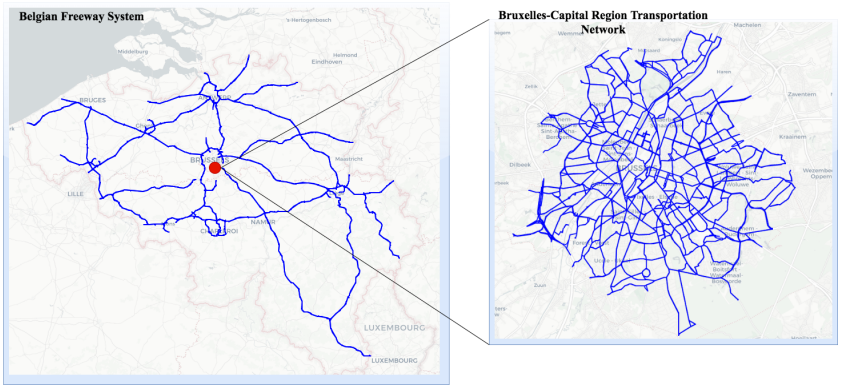
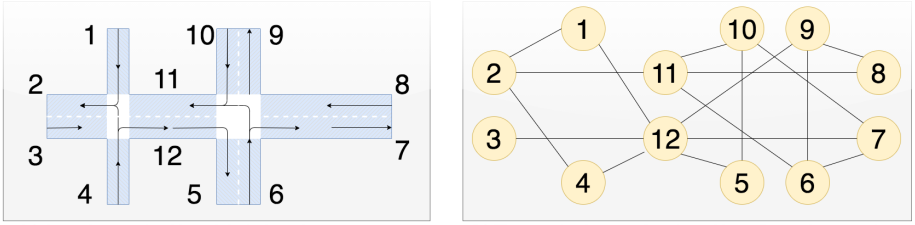


FIGURE 4: Street maps of the Belgian Freeway System and the Bruxelles-Capital Region road network.



(a) A road network with 12 labeled road segments

(b) The graph of 12 road nodes with undirected connections

FIGURE 5: Road segment graph construction from OBU data set. In this type of graph, the road segment represents the node and two connected segments have an edge.

TABLE 1: Types of road network for OBU data.

Road Network	Belgium	Bruxelles Region
Context	Freeway	Urban
Types of Streets	Freeways	Freeways Links, Primary, Secondary and Tertiary Roads
N° of Streets	5795	4524
Resample	30 min.	15 min.

2) the $DaysOfWeek$. Each day of week shows particular pattern of traffic flow;
3) the $Working/WeekEnd\ Days$. There is a clear difference in traffic flow between the working days and the week-end days.

3) Road Segment Graph

To preserve the natural spatial structure of the OBU data in the road networks, we represented them as graphs as shown in Figure 5. We built the graphs for OBU data related to both the freeway and the urban networks. In the graphs a node represents a road segment, the edges indicate the connections between road segments and the features of this node are the average traffic measurements (flow or speed) recorded by all the GPS points on it. As shown in Table 1, the resulting graphs include $N = 5795$ road segments for the Belgian Freeway system and $N = 4524$ for Bruxelles Capital Region road network. To the best of our knowledge, this is the most extensive study in this regard [35]. The code implemented to build the graphs is publicly available [2].

B. MODEL COMPARISONS

To validate the competitiveness of the proposed AST-MTL, various STL, as well as MTL methods, are presented and compared.

In STL scenarios, the models carry out the prediction of traffic flow or traffic speed without joint training the two tasks with a shared module:

1) History Average model (*HA*) [4]: it is a multi-horizon persistence model where observations at the same time slot and the same day of the previous three-weeks seasons are collected and the mean of those observations is returned as forecast;
2) Gated Recurrent Unit model (*GRU*) [21]: see IV-B for details;
3) Long Short-Term Memory Model (*LSTM*) [20]: LSTM shares the same fundamental principles of GRU. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. LSTM and GRU have been seen to perform similarly in various tasks [23];
4) GCN and GRU model (*GCN-GRU*): such model is built only upon the spatio-temporal blocks of Figure 3 and does not include any shared module or attention mechanism.

[2]https://www.kaggle.com/giobbu/un-direct-graph-build

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083412, IEEE Access

Belgian Freeway System - H=12 (6 hours)

$Heads$		$Batch$		$StateSize_{gru,fnn_{1,3}}$	
1	0.079	**32**	**0.074**	25	0.085
2	0.077	64	0.075	50	0.083
4	**0.076**	128	0.081	**150**	**0.079**
8	0.078	256	0.094	250	0.080
$Drop$		Lr		$StateSize_{fnn_2}$	
0.1	**0.077**	0.01	0.147	16	0.087
0.2	0.078	0.005	0.108	32	0.080
0.4	0.078	**0.001**	**0.079**	**64**	**0.078**

Bruxelles Region Urban Network - H=12 (3 hours)

$Heads$		$Batch$		$StateSize_{gru,fnn_{1,3}}$	
1	0.048	32	0.048	25	0.048
2	0.048	**64**	**0.047**	50	0.048
4	**0.047**	128	0.048	**150**	**0.047**
8	0.047	256	0.050	250	0.047
$Drop$		Lr		$StateSize_{fnn_2}$	
0.1	**0.047**	0.01	0.050	16	0.0475
0.2	0.048	0.005	0.048	32	0.0473
0.4	0.048	**0.001**	**0.047**	**64**	**0.0471**

TABLE 2: The hyperparameters are selected from the average MAE result over three splits. According to the rolling origin evaluation, a total of 1728 and 2688 forecast points *per split* is considered respectively for Belgium and Bruxelles Region's road networks. *Heads*: number of heads in the attention mechanism; *Batch*: batch size; $StateSize_{gru,fnn_{1,3}}$: state size of GRU layer and the first and last hidden layer of FNN; *Drop*: dropout layer; *Lr*: learning rate; $StateSize_{fnn_2}$: the second hidden layer of FNN.

In MTL scenarios, the models learn traffic flow and speed jointly:

1) multi-task learning Gated Recurrent Units (*MTL-GRU*) [8]: the model represents the state-of-the-art for MTL traffic forecasting of flow and speed. The model is entirely based on GRU layers for the task-specific and shared representation. A merge layer combines information between tasks and allows to learn a deeper hidden representation. Moreover, a residual connection is used to improve the performance of the model;
2) GCN and GRU with an augmented matrix (*GCN-GRU-amtx*): a unique model based on spatio-temporal blocks learns the past flow and speed observations together in an augmented matrix form. In this case the model shares all parameters across tasks.

C. EXPERIMENTAL SETUP

In the experiments, we consider two months period of OBU data, from the 1st of January to the 28th of February 2019, and forecast both traffic flow and speed for each street segment up to $H = 12$. We partition the dataset into a training set (70%), a validation set (10%) for hyperparameter optimization, and a hold-out set (20%) for testing. We perform hyperparameter optimization by time-series cross-validation [36] and we adopt the rolling origin evaluation [37] to assess the model robustness with respect to outliers or drifts. Table 2 summarizes the results of the hyperparameters optimization for AST-MTL model. The AST-MTL model is trained for 250 epochs by minimizing the mean absolute loss function with the Adam optimizer. The same dataset partition, validation criteria, and range of hyperparameter values apply to all considered approaches. The details of the complete study are available for reproducibility purposes in the GitHub repository [3].

D. PERFORMANCE METRICS

To measure the forecasting accuracy of the competing forecasting approaches we use the scale-dependent Root Mean Squared Error, $RMSE$, and Mean Absolute Error, MAE. For multiple-step ahead forecasts, they are defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{t}\sum_{h=1}^{H}(e_{i,h})^2}{nH}} \tag{13}$$

$$MAE = \frac{\sum_{i=1}^{t}\sum_{h=1}^{H}|e_{i,h}|}{nH} \tag{14}$$

where $e_{i,h}$ is the error of the forecast for period i and forecast horizon h.

Additionally, we report the Mean Absolute Scaled Error, (MASE) proposed in [38]. Such scale-independent metric is particularly suitable for freight transport in the urban context where the traffic flow and speed may be extremely heterogeneous over the road network because of volatility. For seasonal time-series, it is defined as:

$$\text{MASE} = \frac{\text{MAE}}{\text{MAE}_{in-sample,seasonal}} \tag{15}$$

where $\text{MAE}_{in-sample,seasonal}$ is the *training MAE* from a simple one-week seasonal naïve method [38]. When

[3] https://github.com/giobbu/AST-MTL

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083412, IEEE Access

TABLE 3: Average RMSE, MAE and MASE on OBU datasets. Percentages in brackets reflect the gain in the loss versus AST-MTL, with AST-MTL outperforming competing methods across all experiments.

Belgian Freeway System

	Flow			Speed		
	AvgRMSE	***AvgMAE***	***AvgMASE***	***AvgRMSE***	***AvgMAE***	***AvgMASE***
HA	5.88 *(+9.5%)*	3.68 *(+9.2%)*	0.81 *(+14.8%)*	18.38 *(+14.5%)*	10.89 *(+26.8%)*	0.98 *(+22.5%)*
STL_GRU	5.62 *(+4.6%)*	3.59 *(+6.5%)*	0.76 *(+7.04%)*	16.39 *(+2.1%)*	9.01 *(+5.0%)*	0.84 *(+5%)*
STL_LSTM	6.18 *(+15.1%)*	3.85 *(+14.2%)*	0.82 *(+15.5%)*	16.26 *(+1.3%)*	8.69 *(+1.2%)*	0.81 *(+1.25%)*
STL_GCN_GRU	5.53 *(+3%)*	3.56 *(+5.6%)*	0.76 *(+7.04%)*	16.1 *(+0.3%)*	8.87 *(+3.3%)*	0.82 *(+2.5%)*
MTL_GRU	5.67 *(+5.6%)*	3.61 *(+7.1%)*	0.77 *(+8.5%)*	16.16 *(+0.7%)*	8.96 *(+4.3%)*	0.83 *(+3.75%)*
GCN_GRU_amtx	5.99 *(+11.6%)*	3.82 *(+13.4%)*	0.82 *(+15.5%)*	16.18 *(+0.8%)*	8.82 *(+2.7%)*	0.82 *(+2.5%)*
AST_MTL	**5.37**	**3.37**	**0.71**	**16.05**	**8.59**	**0.80**

Bruxelles Region Urban Network

	Flow			Speed		
	AvgRMSE	***AvgMAE***	***AvgMASE***	***AvgRMSE***	***AvgMAE***	***AvgMASE***
HA	0.99 *(+30.2%)*	0.53 *(+32.5%)*	0.89 *(+32.8%)*	11.66 *(+31%)*	5.57 *(+43.6%)*	0.90 *(+45.2%)*
STL_GRU	0.79 *(+4%)*	0.42 *(+5%)*	0.70 *(+4.5%)*	9.06 *(+1.8%)*	4.0 *(+3.1%)*	0.64 *(+3.2%)*
STL_LSTM	0.79 *(+4%)*	0.41 *(+2.5%)*	0.69 *(+3%)*	9.05 *(+1.7%)*	3.92 *(+1.3%)*	0.63 *(+1.6%)*
STL_GCN_GRU	0.78 *(+2.6%)*	0.41 *(+2.5%)*	0.69 *(+3%)*	9.02 *(+1.4%)*	3.92 *(+1.3%)*	0.63 *(+1.6%)*
MTL_GRU	0.79 *(+4%)*	0.42 *(+5%)*	0.70 *(+4.5%)*	9.06 *(+1.8%)*	3.99 *(+2.8%)*	0.64 *(+1.6%)*
GCN_GRU_amtx	0.78 (+2.6%)	0.42 *(+5%)*	0.70 *(+4.5%)*	9.04 *(+1.6%)*	3.96 *(+2.1%)*	0.64 *(+1.6%)*
AST_MTL	**0.76**	**0.40**	**0.67**	**8.90**	**3.88**	**0.62**

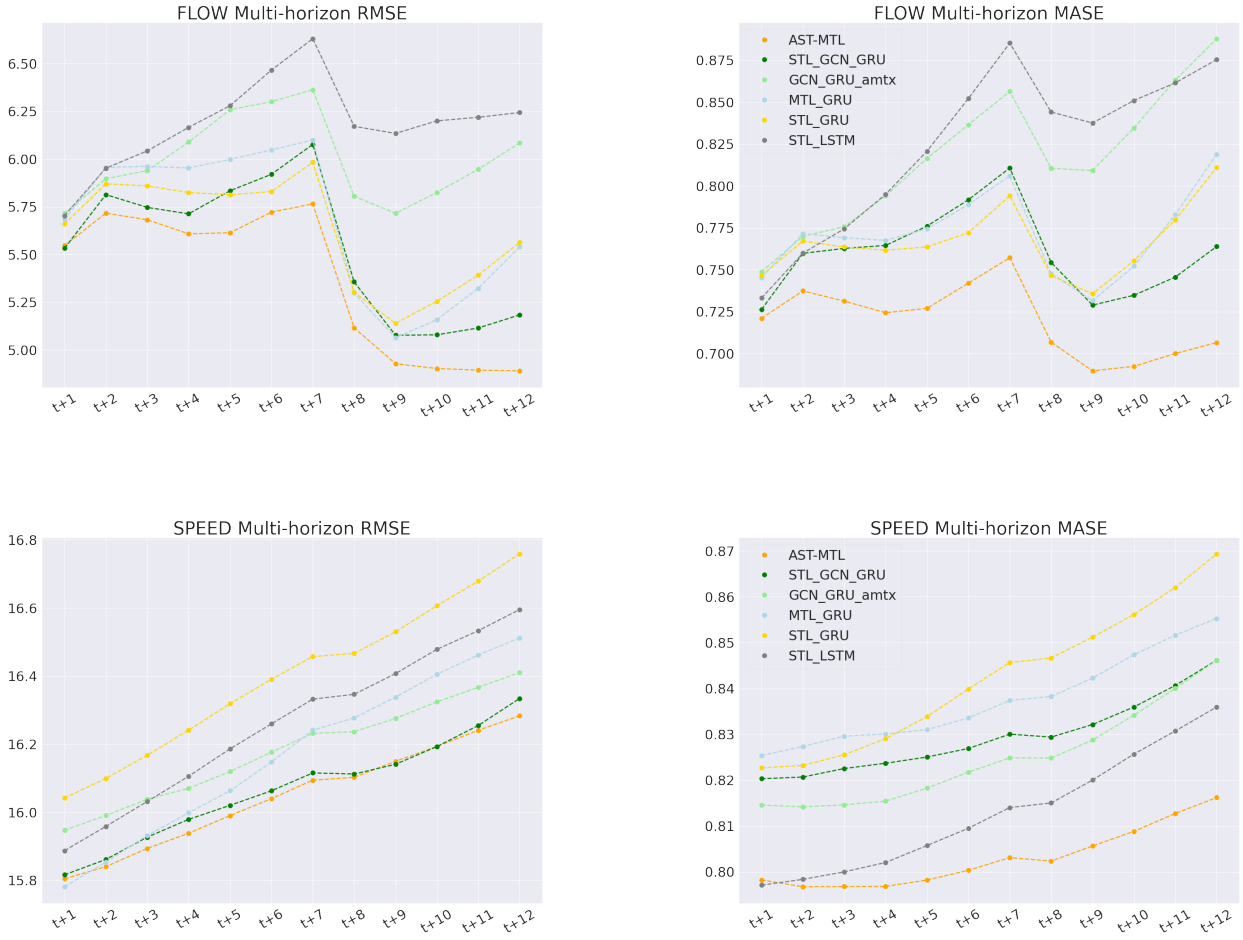


FIGURE 6: Results for Belgian Freeway System. The History Average (HA) model results are not shown as significantly worse.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083412, IEEE Access

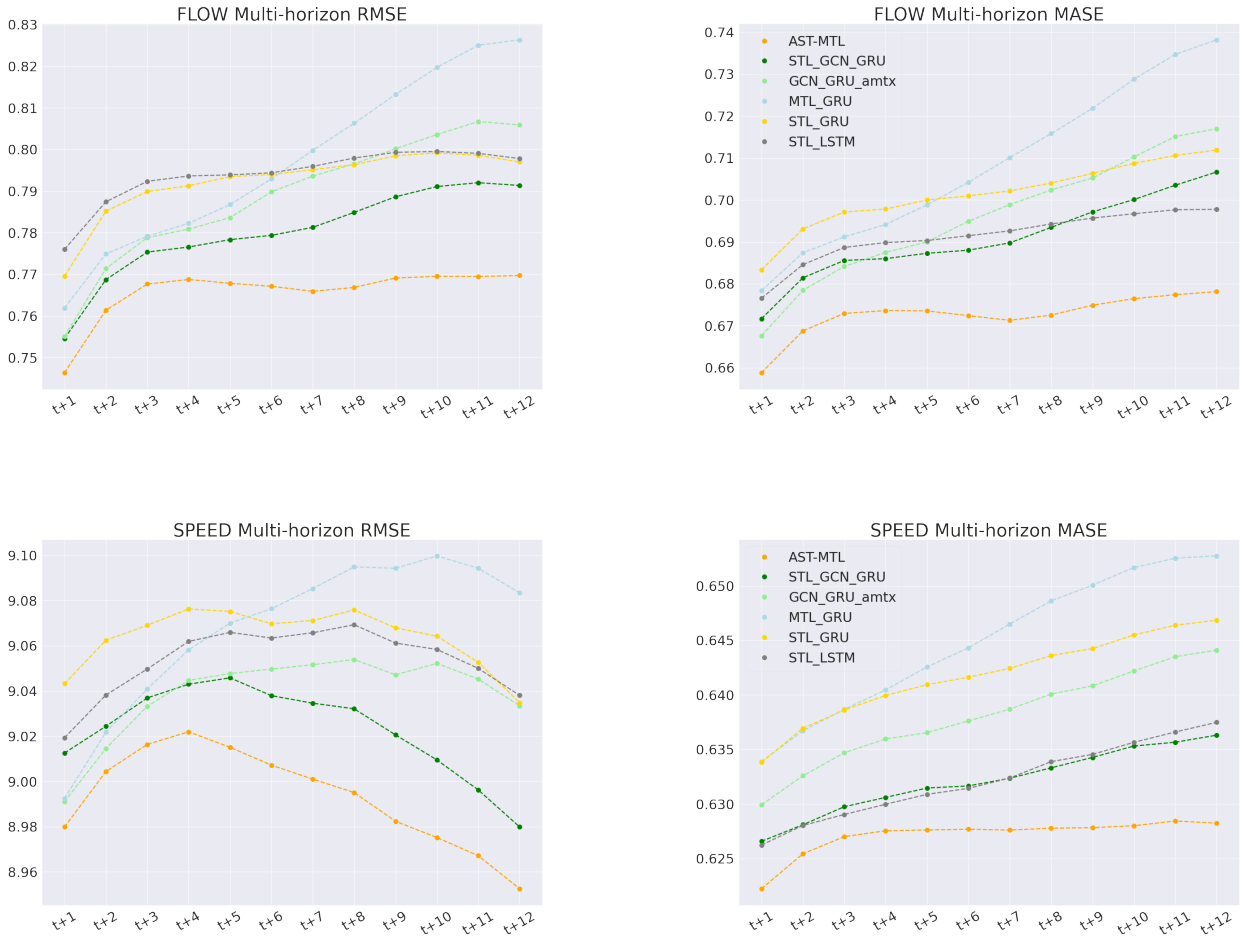


FIGURE 7: Results for Bruxelles Region Urban Network. See Notes Figure 6.

MASE<1, the proposed model performs, on average, better than the naïve method.

VI. RESULTS AND DISCUSSION

In this section, we present the results of the proposed model and compare them with the other counterparts for the Belgian freeway system and the road network of the Bruxelles-Capital Region.

Table 3 shows the benefits of AST-MTL architecture with the general multi-horizon traffic forecasting problem, where it achieves the best accuracy for both traffic flow and speed prediction. We notice that the model is less effective at forecasting traffic conditions of urban roads than freeways due to the complex traffic conditions governing urban networks [1]. Further, we observe the task of traffic flow presents better results compared to the one of speed. As stated by *Crawshaw* [17], this probably because the process of learning one task can sometimes lead to a synergistic or antagonistic effect on the capacity of learning the other.

In Figures 6-7 the average value of *RMSE* (scale-dependent) and *MASE* (scale-independent) are presented for each forecast horizon (the same conclusions can be drawn for *MAE*). The figures further highlight the competitiveness of AST-MTL for both freeway and urban network. In particular, the results show the advantage of applying an attention-based MTL strategy for multiple steps ahead forecasting and especially for long-range forecast horizons.

Furthermore, we remark the STL-GCN-GRU model performs the best among other methods. This proves the importance of modeling both the spatial and temporal components of traffic conditions.

Finally, concerning the remaining models, the results present high variability according to the context of forecasting and performance metric. In particular, the MTL-GRU model proposed by [8] seems to not be particularly effective for multi-horizon traffic forecasting. Since the model is designed to predict the traffic conditions for a single horizon, the results rapidly deteriorate in the long-range forecasting.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083412, IEEE Access

	Belgium	*Bruxelles Region*
	Training Time (min)	**Training Time (min)**
STL_GRU	~ 11	~ 24
STL_LSTM	~ 12	~ 25
STL_GCN_GRU	~ 16	~ 32
MTL_GRU	~ 15	~ 19
GCN_GRU_amtx	~ 15	~ 24
AST-MTL	**~ 18**	**~ 39**

TABLE 4: Run-Time Analysis.

1) Run-time Analysis

The training of deep learning architectures required one NVIDIA TESLA P100 GPU, two Intel(R) Xeon(R) CPU@2.30GHz, and 13 Gigabytes of RAM (freely accessible in Google[45]).

Table 4 reports the training time for different models. From the results, it appears that the complex structure of the proposed model requires a higher computational cost than the other approaches. However, we deem that for traffic prediction applications the computational overload of the AST-MTL model is acceptable since the training is generally carried out in an offline manner where considerable computational resources are available [39]. Moreover, when it comes to real-time prediction, AST-MTL can predict the traffic condition in a matter of seconds.

2) Friedman & post-hoc Nemenyi tests

We use the *Friedman test* to verify whether the differences in performance distributions of different methods are significant. Since there is evidence of this (p-value is 0.000) for traffic flow and speed and for both the freeway and urban context, we proceed to apply the *post-hoc Nemenyi test*. Figures 8-9 present the results at 5% significance level for both traffic variables and contexts according to $RMSE$ (the same conclusions can be drawn for MAE and $MASE$). For each method, the mean rank is provided with the lowest indicating the most accurate one. A horizontal line connects methods in which there is not adequate evidence to suggest statistically significant differences (i.e., the differences of the mean ranks are lower than the critical distance). Following the results in Table 3, the AST-MTL significantly outperforms all benchmarks.

3) Sensitivity Analysis

To assess the benefit of each component for the proposed architecture, we perform a sensitivity analysis. We remove one component at a time while maintaining the MTL structure, and we compute the percentage increase loss versus the original AST-MTL architecture. For the sake of the analysis, we account for those components that are new to the MTL-GRU proposed by [8]:

- **FNN**: this component is responsible to learn the hidden representation shared across related tasks;
- **Attention** : the attention mechanism identifies the salient portions of task-specific information by taking into account the task shared representation;
- **GCN**: the stacked GCN layers capture the spatial information hidden in the observations.

Figures 10-11 display how the contribution of each component to the AST-MTL performance varies according to the

[4]https://www.kaggle.com/docs/efficient-gpu-usage
[5]https://research.google.com/colaboratory/faq.html

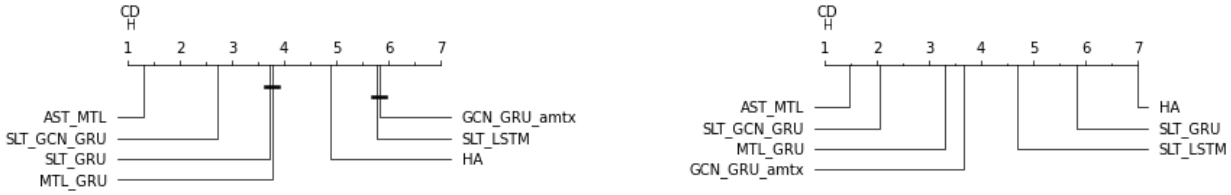


FIGURE 8: Nemenyi tests for Belgian Freeway System: the results of traffic flow (left-side) and speed (right-side) at 5% significance level. The critical distance, CD, is equal to 0.102. A total of $N_{Test} \times N_H = 7788$ forecasting points are tested, where N_{Test} is the number of observations in the testing set while N_H the forecast horizons.

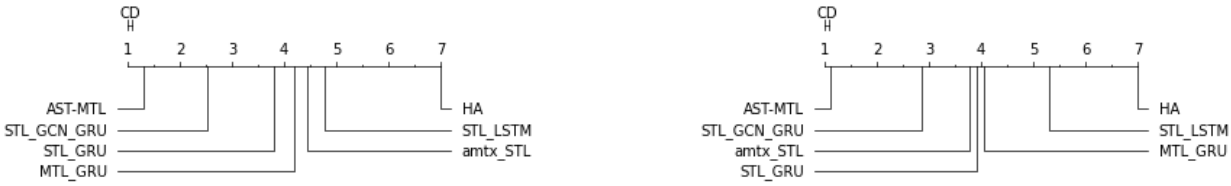


FIGURE 9: Nemenyi tests for the Bruxelles-Capital Region: the results of traffic flow (left-side) and speed (right-side) at 5% significance level. The critical distance, CD, is equal to 0.72. A total of $N_{Test} \times N_H = 15851$ forecasting points are tested.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083412, IEEE Access

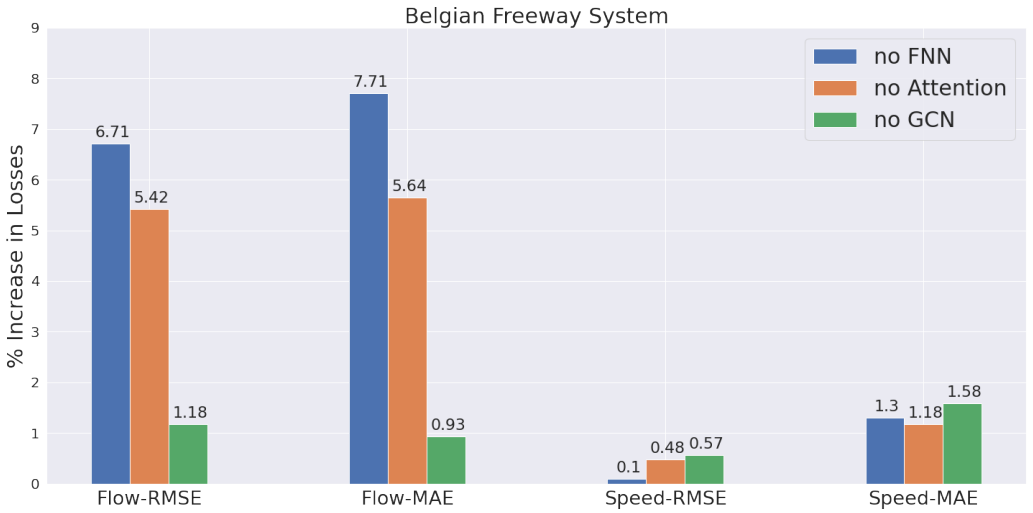


FIGURE 10: RMSE and MAE percentage increase for AST-MTL in the context of freeway traffic prediction.

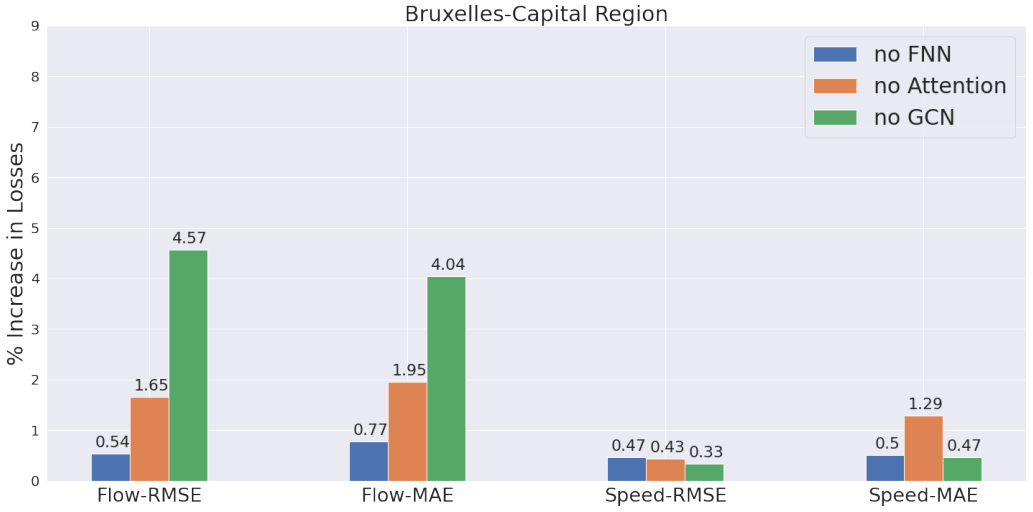


FIGURE 11: RMSE and MAE percentage increase for AST-MTL in the context of urban traffic prediction.

learning task and the type of road network taken into account.

- *Learning Task*: as highlighted in Section VI, the process of learning multiple tasks concurrently is not trivial. The figures clearly show how the components favour the learning of traffic flow at the expense of speed.
- *Type of road network*: considering the traffic flow, the components in charge of capturing the commonalities across tasks (*FNN* and *Attention*) are the most important for the AST-MTL performance in the freeway type of network (Figure 10). This is not valid in the urban context, where the performance of the model mainly relies on the *GCN* layers (Figure 11). In this case, the ability to learn the spatial component of traffic is fundamental when predicting the traffic conditions in road networks characterized by a complex morphology [1] (Figure 4).

VII. CONCLUSION

We introduce AST-MTL, a novel attention-based spatio-temporal multi-task learning model to perform traffic flow and speed forecasting. We validate the methodology on GPS data from the Belgian freeway system and the urban road network of the Bruxelles-Capital Region. The experimental results show AST-MTL achieves the best performance compared with other state-of-the-art deep learning approaches. In particular, the following observations have been highlighted: (1) the attention-based MTL model presents robust results for the multi-horizon forecasting for both traffic flow and traffic speed outperforming [8] and other counterparts, (2) the components of the proposed architecture contribute differently according to the type of road network we perform traffic predictions, and (3) the model does not explicitly optimize the synergy between tasks. To conclude, future research will address the problem of selecting the proper method to

explicitly optimize the learning of tasks. This aspect, together with the definition of the MTL architecture here addressed, is a critical challenge in Multi-Task Learning [17].

REFERENCES

[1] I. Lana, J. Del Ser, M. Velez, and E. I. Vlahogianni, "Road traffic forecasting: Recent advances and new challenges," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 93–109, 2018.

[2] J. S. Angarita-Zapata, A. D. Masegosa, and I. Triguero, "A taxonomy of traffic forecasting regression problems from a supervised learning perspective," *IEEE Access*, vol. 7, pp. 68 185–68 205, 2019.

[3] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2014.

[4] J. Liu and W. Guan, "A summary of traffic flow forecasting methods [j]," *Journal of Highway and Transportation Research and Development*, vol. 3, pp. 82–85, 2004.

[5] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction," *arXiv preprint arXiv:1801.02143*, 2018.

[6] A. Ermagun and D. Levinson, "Spatiotemporal short-term traffic forecasting using the network weight matrix and systematic detrending," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 38–52, 2019.

[7] J. Tang, F. Liu, Y. Zou, W. Zhang, and Y. Wang, "An improved fuzzy neural network for traffic speed prediction considering periodic characteristic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2340–2350, 2017.

[8] K. Zhang, L. Wu, Z. Zhu, and J. Deng, "A multitask learning model for traffic flow and speed forecasting," *IEEE Access*, vol. 8, pp. 80 707–80 715, 2020.

[9] R. Fu, Z. Zhang, and L. Li, "Using lstm and gru neural network methods for traffic flow prediction," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. IEEE, 2016, pp. 324–328.

[10] X. Cao, Y. Zhong, Y. Zhou, J. Wang, C. Zhu, and W. Zhang, "Interactive temporal recurrent convolution network for traffic prediction in data centers," *IEEE Access*, vol. 6, pp. 5276–5289, 2017.

[11] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[12] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

[13] F. Jin and S. Sun, "Neural network multitask learning for traffic flow forecasting," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 1897–1901.

[14] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: deep belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.

[15] K. Zhang, L. Zheng, Z. Liu, and N. Jia, "A deep learning based multitask model for network-wide traffic speed prediction," *Neurocomputing*, vol. 396, pp. 438–450, 2020.

[16] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[17] M. Crawshaw, "Multi-task learning with deep neural networks: A survey," *arXiv preprint arXiv:2009.09796*, 2020.

[18] T. Gong, T. Lee, C. Stephenson, V. Renduchintala, S. Padhy, A. Ndirango, G. Keskin, and O. H. Elibol, "A comparison of loss weighting strategies for multi task learning in deep neural networks," *IEEE Access*, vol. 7, pp. 141 627–141 632, 2019.

[19] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[22] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[23] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[25] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3156–3164.

[26] S. O. Arik and T. Pfister, "Tabnet: Attentive interpretable tabular learning," *arXiv preprint arXiv:1908.07442*, 2019.

[27] E. Choi, M. T. Bahadori, J. A. Kulas, A. Schuetz, W. F. Stewart, and J. Sun, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," *arXiv preprint arXiv:1608.05745*, 2016.

[28] A. Alaa and M. van der Schaar, "Attentive state-space modeling of disease progression," 2019.

[29] B. Lim, S. O. Arik, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *arXiv preprint arXiv:1912.09363*, 2019.

[30] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu, "Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting," *Transactions in GIS*, vol. 24, no. 3, pp. 736–755, 2020.

[31] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *arXiv preprint arXiv:1907.00235*, 2019.

[32] G. Buroni, Y.-A. Le Borgne, G. Bontempi, and K. Determe, "On-board-unit data: A big data platform for scalable storage and processing," in *2018 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech)*. IEEE, 2018, pp. 1–5.

[33] G. Buroni, G. Bontempi, and K. Determe, "A tutorial on network-wide multi-horizon traffic forecasting with deep learning," 2021.

[34] G. Boeing, "Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks," *Computers, Environment and Urban Systems*, vol. 65, pp. 126–139, 2017.

[35] J. Ye, J. Zhao, K. Ye, and C. Xu, "How to build a graph-based deep learning architecture in traffic domain: A survey," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[36] R. J. Hyndman, "Measuring forecast accuracy," *Business forecasting: Practical problems and solutions*, pp. 177–183, 2014.

[37] L. J. Tashman, "Out-of-sample tests of forecasting accuracy: an analysis and review," *International journal of forecasting*, vol. 16, no. 4, pp. 437–450, 2000.

[38] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International journal of forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[39] C. Zheng, X. Fan, C. Wen, L. Chen, C. Wang, and J. Li, "Deepstd: Mining spatio-temporal disturbances of multiple context factors for citywide traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3744–3755, 2019.

GIOVANNI BURONI received the M.Sc. degree in engineering from the University of Bologna, Italy in 2016. He is currently a PhD student at the Machine Learning Group, ULB. His research interests include intelligent transportation systems, online predictions, scalable machine learning, and deep learning.

BERTRAND LEBICHOT received his M.Sc. degree in engineering from the Université catholique de Louvain (UCL, Belgium) in 2011. He later obtained his Ph.D. in the same university in 2018. He worked as a Postdoctoral Researcher at the Université Libre de Bruxelles (ULB, Belgium) and is currently a Research Associate with the University of Luxembourg. He is also a part-time lecturer at UCL. His research interests include graph mining, deep learning, and Fintech applications.

GIANLUCA BONTEMPI is Full Professor in the Computer Science Department at the Université Libre de Bruxelles (ULB), Brussels, Belgium, co-head of the ULB Machine Learning Group (mlg.ulb.ac.be). He has been Director of (IB)2, the ULB/VUB Interuniversity Institute of Bioinformatics in Brussels (ibsquare.be) in 2013-17. His main research interests are big data mining, machine learning, bioinformatics, causal inference, predictive modeling and their application to complex tasks in engineering (time-series forecasting, fraud detection) and life science (network inference, gene signature extraction). He was Marie Curie fellow researcher, he was awarded in two international data analysis competitions and he took part to many research projects in collaboration with universities and private companies all over Europe. He is author of more than 250 scientific publications and his H-number is 59. He is Belgian (French Community) national contact point of the CLAIRE network, co-leader of the CLAIRE COVID19 Task Force, associate editor of the International Journal of Forecasting and IEEE Senior Member. He is also co-author of several open-source software packages for bioinformatics, data mining and prediction.