Large Language Models for Wireless Cellular Traffic Prediction: A Multi-timespan Approach

Mohammad Hossein Shokouhi and Vincent W.S. Wong
Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada
email: {mhshokouhi, vincentw}@ece.ubc.ca

***Abstract*—Wireless cellular traffic prediction is essential for efficient network management and monitoring, yet it is a challenging task due to the spatial and temporal characteristics of traffic. Recently, machine learning based traffic prediction algorithms have been proposed in the literature. However, these algorithms lack good generalization ability as they cannot adapt to frequent changes in traffic distribution typically encountered in wireless networks. In this paper, we propose a traffic prediction algorithm using large language models (LLMs). We first analyze the temporal characteristics of traffic and identify those timespans in the historical traffic information which are important for traffic prediction. We use a clustering algorithm to identify cells with similar traffic patterns. To predict the traffic in a cell, we incorporate the multi-timespan historical traffic information of the cell as well as those cells with similar traffic patterns into natural language sentences and provide them as input to the LLM. Using our proposed framework, we fine-tune three popular LLMs (BART, BigBird, and PEGASUS) on the traffic prediction task. Experimental results show that our proposed LLM framework outperforms a state-of-the-art graph neural network (GNN) baseline and achieves up to 12.32% improvement in terms of the mean absolute error (MAE). Moreover, the proposed LLM framework has excellent generalization ability under the zero-shot setting, reducing the MAE by up to 46.84% compared to the baseline. The ablation studies reveal that providing information from multiple timespans to the model reduces the MAE by up to 15.05% compared to only providing information from the most recent timespan.**

I. Introduction

The current fifth-generation (5G) wireless networks support different mobile applications with diverse quality of service requirements, such as video streaming, virtual reality, and cloud gaming. In order to accommodate these applications and meet their stringent requirements, it is crucial for the network operators to predict wireless traffic so that they can provision the radio and core network resources efficiently.

Wireless cellular traffic depends on various factors, including the time of the day (e.g., weekday vs. weekend, peak hour vs. off-peak hour), special events, public holidays, and population density. Over the past few years, there have been different types of data driven or machine learning based traffic prediction algorithms proposed in the literature. In [1], Wang *et al.* proposed a traffic prediction algorithm using long short-term memory (LSTM) model. In [2], the authors proposed a traffic prediction algorithm using the attention modules and convolutional LSTM [3] to capture the spatial-temporal dependencies. In [4], a Bayesian meta-learning algorithm is proposed to predict wireless traffic in different regions. In [5], an adaptive graph convolutional recurrent network (AGCRN) is proposed which can capture the spatial and temporal correlations. In [6], a dynamic Bernstein graph recurrent network is proposed for wireless traffic prediction.

The generalization ability is an important feature in machine learning based traffic prediction models. If a model can generalize well, it can make accurate predictions even when the model has been trained using a relatively small dataset, thus saving the cost for training and data collection. Furthermore, traffic distributions may change in wireless cellular networks over time due to changes in user behavior or network topology (e.g., new base stations being deployed). A model with good generalization ability can adapt to these changes. A common approach to determine the generalization ability of a model is to evaluate its prediction performance on a set of cells that were not included in the training process, known as zero-shot evaluation. Most of the existing traffic prediction algorithms do not demonstrate good zero-shot performance.

In recent years, there has been significant growth in pre-trained large language models (LLMs), such as GPT-3 [7]. These models are trained on vast text corpora and demonstrate exceptional generalization abilities [8]. The reason is that the self-attention modules in LLMs can perform data-independent operations that are similar to principal component analysis (PCA) over the input patterns. This enables them to serve as universal models applicable to various downstream tasks [9]. Notably, some recent studies have utilized LLMs for time-series forecasting. In [8], the numerical inputs and outputs are transformed into natural language and different LLMs are fine-tuned on the time series forecasting task. In [9], the LLM's self-attention and feedforward layers are frozen, while the positional embedding and layer normalization layers are fine-tuned on the time series forecasting task. However, these models are not applicable to wireless cellular traffic prediction since they consider either a single time series [8] or multiple uncorrelated time series [9]. The spatial correlations of the traffic patterns between different cells were not considered in those LLM frameworks.

Inspired by the strong generalization ability of LLMs, in this paper we utilize LLMs for wireless cellular traffic prediction. Since the computational complexity of LLMs increases linearly with the size of the input, feeding the entire traffic history to the model is impractical. On the other hand, only including the information from the most recent time steps may not always lead to accurate prediction results. Thus, we

identify the timespans in traffic history that are most important for traffic prediction. Then, we provide the information from these timespans to the model to reduce the input length while increasing the efficiency. Furthermore, in order to leverage the spatial correlation between cells, our model's input includes the traffic history of cells with similar traffic patterns as auxiliary information to refine the predictions. The main contributions of this paper are summarized as follows:

- We leverage the capabilities of LLMs for wireless cellular traffic prediction. We model the traffic prediction problem as a natural language processing (NLP) task and design input sentences, called prompts, that encapsulate the traffic history and inquire about future traffic. Subsequently, the model generates responses, which contain the predicted traffic values. We fine-tune three popular LLMs (BART [10], BigBird [11], and PEGASUS [12]) and assess their performance on the traffic prediction task.
- Instead of only using the information from the recent time steps, we analyze the temporal autocorrelation of traffic to identify timespans in historical traffic information that are important for traffic prediction. We include the information from these timespans in the input prompts.
- In order to exploit the spatial correlation between cells, we train an autoencoder to obtain a low-dimensional embedding for the traffic in each cell. Then, we invoke a clustering algorithm to identify cells with similar traffic patterns. When predicting the future traffic of each cell, we provide the traffic history of cells with similar traffic patterns as auxiliary information to refine the predictions.
- Simulation results show that our proposed LLM framework outperforms AGCRN [5], reducing the mean absolute error (MAE) by up to 12.32%. Furthermore, it shows good generalization ability, reducing the MAE by up to 46.84% under the zero-shot setting compared to the baseline.
- Results from the ablation studies show the importance of using multiple timespans for accurate traffic prediction. Including multi-timespan information can reduce the MAE by up to 15.05% compared to only considering the information from the most recent timespan.

The rest of this paper is organized as follows. In Section II, we present the traffic prediction problem and our proposed LLM framework. In Section III, we evaluate the performance between our proposed LLM framework and a baseline scheme, assess its generalization ability, and present results on the ablation studies. Conclusion is given in Section IV. *Notations*: We use $\mathbb{N}$ and $\mathbb{N}^+$ to denote the set of non-negative integers and positive integers, respectively.

II. System Model

In this section, we first present the wireless cellular traffic prediction problem. Then, we explain the importance of considering multiple timespans. We also show the correlation between the traffic patterns of neighboring cells and discuss how this information can be exploited for traffic prediction. We then introduce the input prompts for LLMs that include the information from multiple timespans and inquire about future traffic values in a natural language manner.

A. Traffic Prediction Problem

Consider a geographical area is divided into N cells, each served by a base station. Let $\mathcal{N} = \{1, \ldots, N\}$ denote the set of cells. Let $\mathcal{T} = \{1, \ldots, T\}$ denote the set of time steps measured in hours, where T is the total number of time steps. We denote the traffic volume of cell $n \in \mathcal{N}$ during time step $t \in \mathcal{T}$ as d_t^n. Let $\boldsymbol{d}^n = (d_1^n, \ldots, d_T^n)$ denote the traffic vector of cell n over all T time steps and $\boldsymbol{d}_t = \left(d_t^1, \ldots, d_t^N\right)$ denote the traffic vector observed during time step t.

The conventional traffic prediction problem is formulated as predicting the traffic volume during the next Q time steps based on the historical traffic information from the past P time steps. In other words, the objective is to maximize the conditional probability of future traffic given the historical traffic information:

$$\boldsymbol{d}_{t:t+Q}^{\star} = \underset{\boldsymbol{d}_{t:t+Q}}{\arg\max} \; p\left(\boldsymbol{d}_{t:t+Q} \mid \boldsymbol{d}_{t-P:t}\right). \tag{1}$$

We will show in Section II-B that, instead of only using the information from the most recent P time steps, it is beneficial to use the information from other timespans as well.

B. Temporal Dependencies

We analyze the temporal dependencies in traffic data to identify the timespans in traffic history that are informative for traffic prediction. The autocorrelation as a function of time lag l is widely adopted in the literature for temporal dependency analysis. The temporal autocorrelation for traffic vector $\boldsymbol{d}^n$ of cell n is as follows

$$r^n(l) = \frac{\sum_{t=1}^{T-l}\left(d_t^n - \bar{d}^n\right)\left(d_{t+l}^n - \bar{d}^n\right)}{\sum_{t=1}^{T}\left(d_t^n - \bar{d}^n\right)^2}, \quad 0 \le l < T, \tag{2}$$

where $\bar{d}^n$ denotes the mean value of traffic vector $\boldsymbol{d}^n$ in cell n. Here, $r^n(l)$ is in the range of $[-1, 1]$ and shows how much the current traffic is correlated to the traffic during l time steps ago. Fig. 2 shows the temporal autocorrelation of aggregated Internet traffic across all cells in Milan, Italy. It can be observed that the current traffic is highly correlated to the most recent time steps. Thus, we define timespan 0 as follows

$$\mathcal{T}_0 = [t - P_0, t), \tag{3}$$

where t is the current time step and $P_0 \in \mathbb{N}$ is the duration of timespan in hours. It is also evident from Fig. 2 that the subsequent peaks occur in $l = 24, 48, \ldots$, suggesting that the current traffic is similar to the traffic during the same time step in previous days. We define timespan $i \in \mathbb{N}^+$ to include the time steps centered around $t - 24i$ and express it as

$$\mathcal{T}_i = \left[t - 24i - \left\lfloor \frac{P_i}{2} \right\rfloor, t - 24i + \left\lfloor \frac{P_i + 1}{2} \right\rfloor\right), \tag{4}$$

where $P_i \in \mathbb{N}$ is the duration of timespan i in hours. For instance, when P_1 is equal to 3, we have $\mathcal{T}_1 = [t - 25, t - 22)$. Without loss of generality, we only provide the information

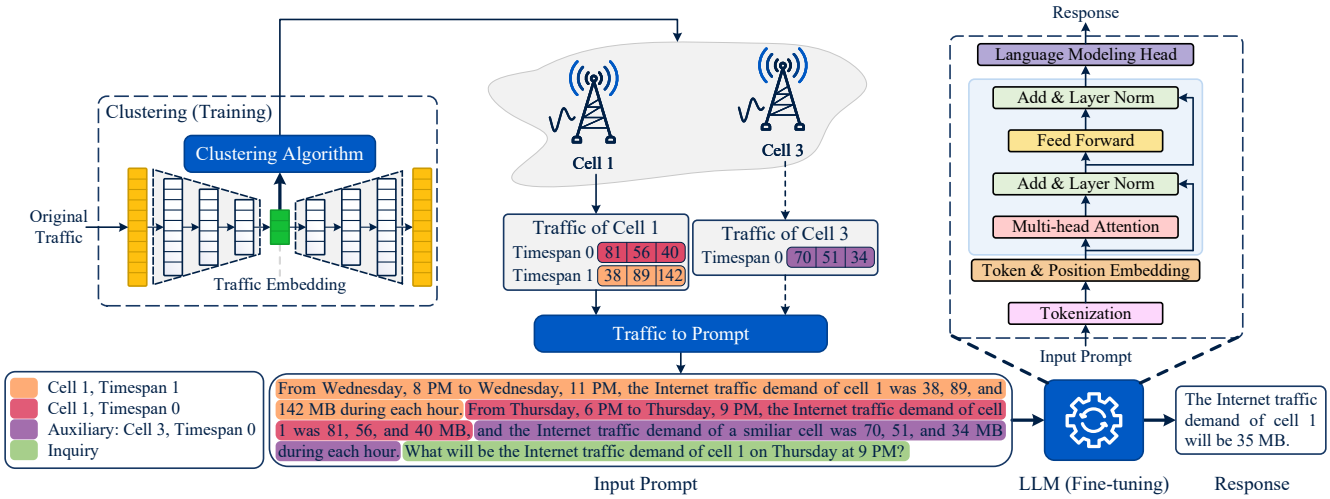


Fig. 1. Illustration of the proposed LLM framework. The autoencoder determines a low-dimensional embedding of traffic history for each cell. The clustering algorithm utilizes these embeddings to identify a set of auxiliary cells for each cell that have similar traffic patterns. To predict the traffic in a cell, the multi-timespan historical traffic information of that cell and its auxiliary cells is incorporated into the prompts and provided as input to the LLM.

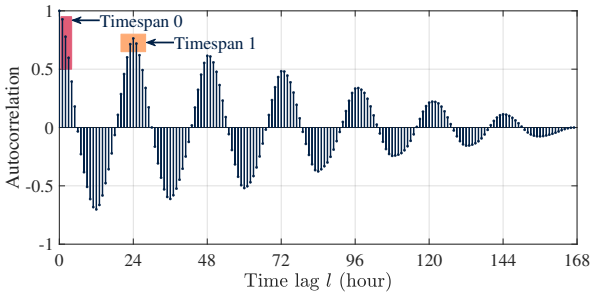


Fig. 2. Temporal autocorrelation of aggregated Internet traffic across all cells in Milan, Italy.

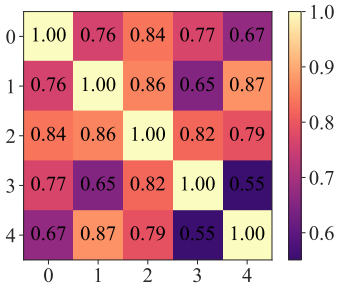


Fig. 3. Spatial correlation between five randomly selected neighboring cells.

from the first two timespans, $\mathcal{T}_0$ and $\mathcal{T}_1$, to the LLM in order to keep the input length to be short. Consequently, in problem (1), the future traffic $\boldsymbol{d}_{t:t+Q}$ will be conditioned on $\boldsymbol{d}_{\mathcal{T}_0}$ and $\boldsymbol{d}_{\mathcal{T}_1}$, which represent the traffic vector during timespans $\mathcal{T}_0$ and $\mathcal{T}_1$, respectively. As an example, suppose we would like to predict the traffic on Monday at 5 PM, and both P_0 and P_1 are set to 3. In this scenario, $\mathcal{T}_0$ represents the timespan from Monday 2 PM to 5 PM, and $\mathcal{T}_1$ represents the timespan from Sunday 4 PM to 7 PM. We gather the historical traffic information from these timespans and provide them to the LLM. In Section II-D, we demonstrate how to incorporate this information into the input prompt of LLMs.

C. Spatial Correlation

We use the Pearson correlation coefficient to measure the spatial correlation between two cells n and n'. It is defined as

$$\rho = \frac{cov\left(\boldsymbol{d}^n, \boldsymbol{d}^{n'}\right)}{\sigma_{\boldsymbol{d}^n}\sigma_{\boldsymbol{d}^{n'}}}, \tag{5}$$

where $cov(.)$ denotes the covariance operation and σ is the standard deviation. The Pearson correlation coefficient is within the range of $[-1, 1]$. Fig. 3 shows the spatial correlation matrix for five randomly selected neighboring cells. It shows that traffic of neighboring cells is highly correlated.

D. Proposed LLM Framework

Most of the existing frameworks on LLMs consider uncorrelated variables, and tasks with spatial correlation between variables are not considered. In this paper, we identify cells with similar traffic patterns and design input prompts that exploit the spatial correlation. To achieve this, in our proposed LLM framework, we first train an autoencoder to encode the traffic history of each cell into a low-dimensional vector. The autoencoder is depicted in Fig. 1. Each of the encoder and decoder modules consists of three fully connected layers with 512, 256, and 128 hidden neurons. Each layer is followed by a rectified linear unit (ReLU) activation function. There are 64 hidden neurons in the middle layer. The output of the middle layer serves as the low-dimensional representation of the traffic, expressed as $\boldsymbol{z}^n = f_{\text{encoder}}(\boldsymbol{d}^n)$. The training objective is to minimize the reconstruction loss defined as

$$\mathcal{L} = \frac{1}{NT}\sum_{n=1}^{N} \left\| \boldsymbol{d}^n - f_{\text{decoder}}(\boldsymbol{z}^n) \right\|_2^2. \tag{6}$$

Next, we invoke a clustering algorithm on the traffic embeddings $\{\boldsymbol{z}^n\}_{n\in\mathcal{N}}$ to identify cells which have similar traffic patterns. Suppose that for each cell, we would like to include auxiliary information from up to S similar cells. We run the

K-means clustering algorithm [2] H times. The algorithm might yield a different clustering result in each run. Let $s_{n,m}$ denote the number of runs in which cells n and m, $n \neq m$, were placed in the same cluster. For each cell n, we identify S cells that have the highest $s_{n,m}, m \in \mathcal{N}$, values. These cells are considered as the auxiliary cells of cell n. This is based on the assumption that cells appearing together in clusters more frequently are more likely to have similar traffic patterns. Furthermore, there might be cells that have unique traffic patterns not correlated with the traffic patterns of any other cells. In order to prevent the algorithm from choosing random auxiliary cells in this case, we set a threshold H_{THR}. An auxiliary cell m is chosen only if $s_{n,m} \geq H_{\text{THR}}$. Next, we show how to provide the multi-timespan and auxiliary information to the LLMs and use them for traffic prediction.

Fig. 1 shows the components of an LLM. The input prompt is first broken into a sequence of tokens. Each token is represented by a high-dimensional one-hot vector. The embedding module determines a dense vector representation (embedding) for each token. The multi-head self-attention module determines the importance of each token in the input sequence relative to the others, enabling the model to capture long-term dependencies in input sequences. The output of self-attention is fed into a feed-forward neural network to generate the output sequence. Layer normalization and residual connections are added after each layer to stabilize the training process. Finally, we use the language modeling head from the *Conditional Generation* class provided by Hugging Face to convert the output embeddings back into one-hot vectors.

Both natural language generation and traffic prediction are sequential tasks that involve sequence-to-sequence generation. The autoregressive generation process of a language model can be expressed as

$$p(y_{1:J}) = \prod_{j=1}^{J} p(y_j \mid y_{<j}), \tag{7}$$

where $p(y_{1:J})$ is the probability of generating the entire sequence $y_{1:J}$, y_j is the token generated in position j, and $y_{<j}$ represents the tokens generated before position j. This expression resembles the traffic prediction problem formulated in problem (1). Both involve generating new tokens based on the previous tokens. Thus, LLMs can be fine-tuned to capture different traffic patterns and predict future traffic. Through fine-tuning, the LLM adapts its pre-learned embeddings to the traffic prediction task, while the self-attention mechanism learns to focus on the most important historical traffic information, and the feed-forward layers learn to predict the traffic based on this information.

In order to fine-tune and use LLMs for traffic prediction, we need to express the inputs and outputs of the traffic prediction task in a natural language format. First, we use a sliding window with a stride of one hour to split the dataset into multiple samples. Each sample of cell n includes information from the P_0 and P_1 time steps of timespans $\mathcal{T}_0$ and $\mathcal{T}_1$, respectively, and Q predictions. For each cell n, we also include information from timespan $\mathcal{T}_0$ of the auxiliary cells. Next, we design input prompts that incorporate the information into natural language sentences and ask for predictions. We use a template for generating prompt-response pairs across all samples. Recent studies have shown that including information such as time-of-day and day-of-week can improve the prediction accuracy [8]. Thus, for each sample, the input prompt includes historical traffic information, time-of-day, day-of-week, cell index, and the question. We generate an individual sentence for each timespan. An example of the input prompt for Internet traffic is given in Fig. 1. The first sentence in the example includes historical traffic information for cell 1 during timespan 1. The second sentence combines information from cell 1 and an auxiliary cell, both during timespan 0. For auxiliary information, rather than specifying the index of the auxiliary cell, we use the term *"similar"* to imply the correlation between the two cells. The final sentence asks about the traffic at the desired time. The response includes the cell index and the prediction. We fine-tune the LLMs on these prompt-response pairs.

Note that our idea of using multiple timespans can be applied to other traffic prediction models, including GNN-based models. However, it typically requires modifying the model architecture and adding extra layers to incorporate information from various timespans. In contrast, when using LLMs, the idea can be implemented by simply updating the input prompt to include the additional information.

III. Performance Evaluation

We use the Telecom Italia Big Data Challenge dataset [13] for performance evaluation. It includes the short message service (SMS), voice call, and Internet traffic data of 10,000 cells organized in a 100×100 grid structure in the city of Milan, Italy. Each cell is a square with the size of 235×235 meters. The traffic is recorded from November 1, 2013 to January 1, 2014 in 10-minute intervals. We aggregate the traffic data of each cell into hourly intervals. Without loss of generality, we randomly select 100 cells from the dataset. We use the data from the first 55 days as our training set and the data from the remaining 6 days as our test set. The prediction horizon Q is set to 1. We choose $P_0 = 3$ and $P_1 = 3$ for the LLMs. For a fair comparison, we set $P_0 = 6$ for the baseline model so that the total duration of the provided information is the same for the LLMs and the baseline. The MAE and root mean squared error (RMSE) metrics are used to measure the prediction performance. The baseline scheme and the LLMs used in our proposed model are as follows:

1) **Adaptive graph convolutional recurrent network (AGCRN) [5]:** This is one of the state-of-the-art GNN-based models. We implement the model using the official AGCRN repository. For fair comparison, we perform hyperparameter tuning to obtain the best results.
2) **LLMs:** We choose three popular LLMs: BigBird [11], PEGASUS [12], and BART [10]. BigBird and PEGASUS have pre-trained model sizes of 2.25 GB and 2.23 GB, respectively, while BART is a lightweight model

TABLE I
PERFORMANCE OF OUR PROPOSED MODEL USING DIFFERENT LLMS AND AGCRN BASELINE SCHEME ON THE TRAFFIC PREDICTION TASK.

Model	SMS traffic		Call traffic		Internet traffic	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
BART	31.43	65.26	17.4	32.28	101.94	162.54
BigBird	29.92	61.05	17.75	33.46	**97.87**	**154.9**
PEGASUS	**29.55**	61.9	**17.15**	**32.08**	99.22	159.96
AGCRN	31.59	**59.51**	19.56	42.04	105.97	156.61
Reduction in %	+6.46	−2.59	+12.32	+23.69	+7.64	+1.09

with a size of 557.8 MB. We use the Hugging Face platform to download the pre-trained models and fine-tune them on the traffic prediction task. We use the tokenizers provided by Hugging Face for each model.

For the autoencoder, we use the Adam optimizer with a learning rate of 0.001 and a weight decay factor of 10^{-8}. We train the autoencoder for 400 epochs with a batch size of 32. For the clustering algorithm, we set $H = 1000$, $H_{\text{THR}} = 500$, and $S = 1$. We use the default hyperparameters recommended by Hugging Face to fine-tune the LLMs. Through experiments, we notice that BART performs the best without normalization, whereas BigBird and PEGASUS achieve better results with max-min normalization. AGCRN achieves its best performance with Z-score normalization. We use normalized traffic data during training to compute the loss and update the weights. During testing, we convert the predicted values to their original scale to determine the true loss.

1) Evaluation Results: Table I shows the evaluation results of different models on the traffic prediction task. It can be observed that BigBird and PEGASUS outperform AGCRN in terms of MAE by up to 6.46%, 12.32%, and 7.64% for SMS, call, and Internet traffic, respectively. The lightweight BART LLM yields comparable performance to AGCRN. To investigate this performance improvement, let us have a closer look at the traffic data in the Milan dataset. Fig. 4 shows the entire Internet traffic history of a randomly selected cell. It can be observed that during the last week, which also constitutes our test set, the traffic values have an abrupt decrease. We found this to be the case for most other cells as well. This is a particularly challenging scenario for most traffic prediction models due to the non-identical distribution of training and test sets. However, LLMs can adapt to this change in the traffic pattern and maintain a good prediction performance due to their better generalization ability. This generalization ability, along with the multi-timespan information, are the main reasons behind the superior performance of LLMs.

Fig. 5 depicts the ground truth and the predictions of AGCRN and our proposed LLM framework using BigBird for SMS, call, and Internet traffic at a randomly selected cell. It can be observed that our proposed LLM framework captures the patterns of the ground truth traffic accurately and outperforms AGCRN in terms of MAE and RMSE for call and Internet traffic. However, it struggles with certain SMS traffic peaks, such as the sharp increase on the final day shown in Fig. 5(a), which leads to a higher RMSE compared to AGCRN

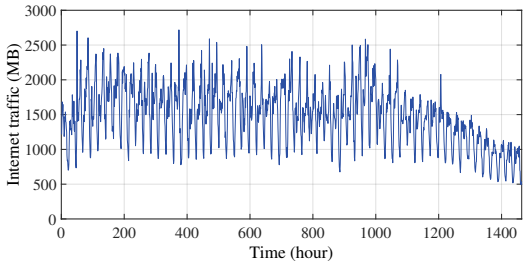


Fig. 4. The entire Internet traffic history of a randomly selected cell.

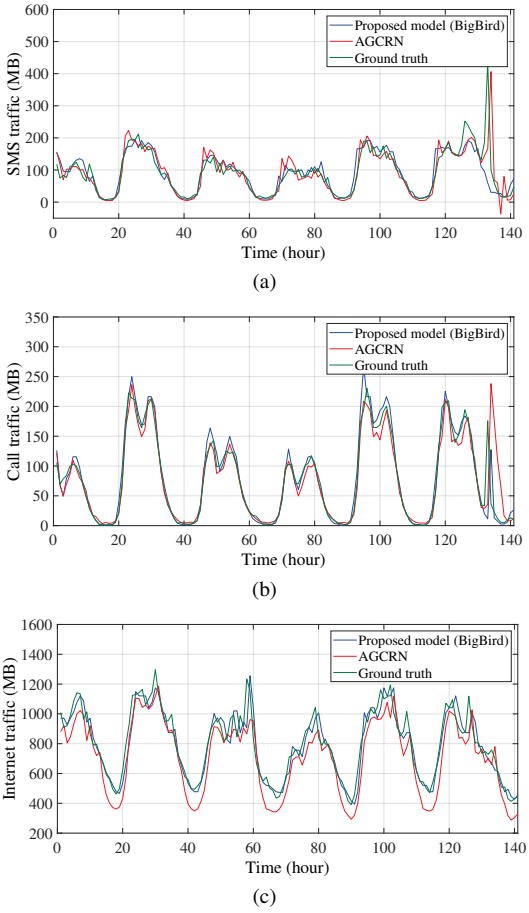


Fig. 5. Ground truth vs. predictions of our proposed model and AGCRN for (a) SMS, (b) call, and (c) Internet traffic at a randomly selected cell.

for SMS traffic. This is because the model leverages historical traffic data across all timespans rather than focusing solely on the most recent period, so it occasionally overlooks abrupt patterns that are confined to a specific timespan.

2) Generalization Ability: We assess the performance of both LLMs and AGCRN in a zero-shot setting to evaluate their generalization ability. In particular, we evaluate their performance on cells that were not included in the training process. To achieve this, we utilize our initial training dataset and select an additional 100 cells to create a test set. The results are shown in Table II. LLMs show a much stronger generalization ability compared to AGCRN, reducing the MAE and RMSE by up to 46.84% and 43.71%, respectively. These results show that LLMs have learned the underlying traffic features instead of memorizing the training data, which

TABLE II
ZERO-SHOT PERFORMANCE OF OUR PROPOSED MODEL USING THREE DIFFERENT TYPES OF LLMS AND THE AGCRN BASELINE SCHEME.

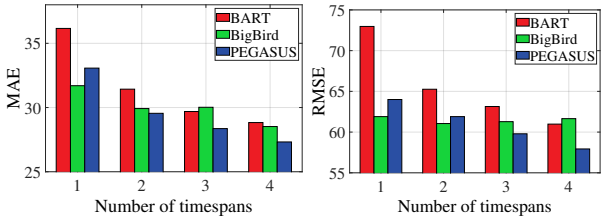
Model	SMS traffic		Call traffic		Internet traffic	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
BART	26.97	58.3	16.59	32.72	99.31	186.39
BigBird	26.56	57.3	16.88	32.79	**86.43**	**137.17**
PEGASUS	**25.51**	**57.07**	**14.7**	**27.78**	88.94	145.38
AGCRN	34.91	58.13	27.65	49.35	160.77	238.24
Reduction in %	26.93	1.82	46.84	43.71	46.24	42.42

leads to a substantial performance improvement over AGCRN.

3) Ablation Study: We perform ablation studies to measure the influence of each component in our proposed model. First, we remove the multi-timespan information from the input prompts, only providing the model with information from timespan $\mathcal{T}_0$. We set $P_0 = 6$ so that the total duration of the provided information remains the same as before. Table III presents the results for SMS traffic. Results show that the MAE performance of BART, BigBird, and PEGASUS LLMs is degraded by 15.05%, 5.95%, and 11.91%, respectively. We notice that including multi-timespan information can significantly improve the prediction accuracy. For further study, Fig. 6 shows the MAE and RMSE of BART, BigBird, and PEGASUS LLMs for SMS traffic when considering different number of timespans. We observe that including information from additional timespans, such as $\mathcal{T}_2$ and $\mathcal{T}_3$, can reduce the MAE and RMSE. We note that the correlation between distant timespans and current traffic tends to decrease, leading to diminishing returns in MAE and RMSE improvement as we include more timespans. Finally, we remove the historical traffic information of auxiliary cells from the input prompts. Results are shown in Table III. Without the auxiliary information, the MAE performance of BART, BigBird, and PEGASUS LLMs is degraded by 0.29%, 1.54%, and 0.85%, respectively.

TABLE III
ABLATION STUDY. LLMS WITH SUFFIXES (W.O. MT) AND (W.O. AU) ARE FINE-TUNED WITHOUT MULTI-TIMESPAN AND AUXILIARY INFORMATION, RESPECTIVELY.

LLM	MAE		RMSE	
	Value	Increase	Value	Increase
BART	31.43	0%	65.26	0%
BART (w.o. MT)	36.16	15.05%	72.97	11.81%
BART (w.o. Au)	31.52	0.29%	65.69	0.66%
BigBird	29.92	0%	61.05	0%
BigBird (w.o. MT)	31.7	5.95%	61.9	1.39%
BigBird (w.o. Au)	30.38	1.54%	62.9	3.03%
PEGASUS	29.55	0%	61.9	0%
PEGASUS (w.o. MT)	33.07	11.91%	64	3.39%
PEGASUS (w.o. Au)	29.8	0.85%	63	1.78%



(a) (b)

Fig. 6. (a) MAE and (b) RMSE of BART, BigBird, and PEGASUS LLMs for different number of considered timespans.

IV. CONCLUSION

In this paper, we modeled wireless cellular traffic prediction as an NLP task and utilized LLMs for cellular traffic prediction. Our proposed model provides LLMs with information from multiple timespans. Additionally, for each cell, we provided auxiliary information from cells with similar traffic patterns to refine the predictions. Experiments demonstrated that our proposed LLM framework outperforms the AGCRN baseline model in terms of MAE and RMSE. For zero-shot performance, results showed that our proposed LLM framework outperforms the baseline due to its generalization ability. Results on ablation studies showed that incorporating multi-timespan information can significantly improve the prediction accuracy. In our current work, we used fixed timespans for all predictions. For future work, we will explore dynamically selecting timespans based on contextual factors such as the day of the week.

REFERENCES

[1] W. Wang, C. Zhou, H. He, W. Wu, W. Zhuang, and X. Shen, "Cellular traffic load prediction with LSTM and Gaussian process regression," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020.

[2] Z. Wang and V. W.S. Wong, "Cellular traffic prediction using deep convolutional neural network with attention mechanism," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Seoul, South Korea, May 2022.

[3] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Montreal, Canada, Dec. 2015.

[4] Z. Wang and V. W.S. Wong, "Bayesian meta-learning for adaptive traffic prediction in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 6, pp. 6620–6633, Jun. 2024.

[5] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2020.

[6] A. Mehrabian, S. Bahrami, and V. W.S. Wong, "A dynamic Bernstein graph recurrent network for wireless cellular traffic prediction," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Rome, Italy, May 2023.

[7] T. Brown, B. Mann *et al.*, "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2020.

[8] H. Xue and F. D. Salim, "PromptCast: A new prompt-based learning paradigm for time series forecasting," *IEEE Trans. Knowl. Data Eng.*, early access, Dec. 2023, doi:10.1109/TKDE.2023.3342137.

[9] T. Zhou, P. Niu, X. Wang, L. Sun, and R. Jin, "One fits all: Power general time series analysis by pretrained LM," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, New Orleans, LA, Dec. 2023.

[10] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Jul. 2020.

[11] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed, "Big bird: Transformers for longer sequences," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2020.

[12] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2020.

[13] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, "A multi-source dataset of urban life in the city of Milan and the province of Trentino," *Scientific data*, vol. 2, no. 1, pp. 1–15, Oct. 2015.