Enhancing Spatiotemporal Networks with xLSTM: A Scalar LSTM Approach for 5G Traffic Forecasting

Zineddine Bettouche, Khalid Ali, Andreas Fischer, Andreas Kassler
Deggendorf Institute of Technology
Dieter-Görlitz-Platz 1, 94469 Deggendorf, Germany
{zineddine.bettouche, khalid.ali, andreas.fischer, andreas.kassler}@th-deg.de

***Abstract*—Accurate spatiotemporal traffic forecasting is vital for optimizing 5G networks. Traditional LSTM models struggle with capturing complex spatiotemporal dependencies, limiting predictive performance. To address this, we propose an enhanced Spatiotemporal Network (STN) integrating Scalar LSTM (sLSTM), a more efficient variant designed to improve temporal modeling while reducing computational complexity. Our dual-path STN processes the input through an sLSTM for sequential feature extraction and a three-layer Conv3D path for spatial feature learning, with both outputs fused in a dedicated fusion layer for enhanced spatiotemporal representation. By incorporating sLSTM, our model stabilizes gradients, accelerates convergence, and enhances accuracy. Experiments on real-world mobile traffic datasets show a 23% MAE reduction over ConvLSTM, with a 30% improvement on unseen data, demonstrating superior generalization for 5G traffic prediction.**

***Index Terms*—5G traffic forecasting, spatiotemporal modeling, deep learning, LSTM, ConvLSTM, xLSTM, scalar-LSTM, AI-driven telecommunications.**

I. Introduction

Accurate spatiotemporal forecasting is essential for optimizing network operations, particularly for traffic prediction, resource allocation, and congestion management. Especially softwarized networks, i.e., networks managed with high dynamicity by software, need proper prediction and forecasts of future traffic demands to adequatly reconfigure network resources in time. However, predicting future values in a time series while incorporating spatial dependencies remains challenging. Traditional Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, while effective for sequential data modeling, struggle with capturing complex spatial correlations and suffer from vanishing gradients.

To address these limitations, Convolutional LSTM (ConvLSTM) networks [1] were introduced, integrating convolutional operations into LSTM architectures to jointly model spatial and temporal patterns. Zhang and Patras [2] extended this concept by developing the Spatiotemporal Network (STN), a dual-path architecture that employs a ConvLSTM branch alongside a fusion layer to enhance mobile traffic prediction. Although these advancements improve spatial modeling, ConvLSTM still suffers from high computational overhead and inefficient gradient flow, limiting its scalability in large-scale 5G traffic forecasting tasks. An alternative approach was proposed by Beck et al. [3] in the form of Scalar LSTM (sLSTM), introduced within the Extended-LSTM (xLSTM) framework. Unlike traditional LSTMs, sLSTM simplifies the gating mechanism using scalar operations, significantly reducing parameter complexity while improving convergence stability and temporal dependency modeling.

In this work, we enhance spatiotemporal traffic forecasting for 5G networks by integrating sLSTM into the STN framework. Our key contributions are:

- sLSTM-Enhanced STN: We incorporate sLSTM into the STN framework, reducing computational overhead and improving predictive performance, achieving a 23% reduction in Mean Absolute Error (MAE) compared to ConvLSTM.
- Attention-based Fusion Mechanisms: We investigate spatial attention, which dynamically re-weights spatial dependencies, and transformer-based fusion, which enhances long-range correlations. While transformer-based fusion improves feature integration, we analyze its computational trade-offs for real-world forecasting.
- Extensive Evaluation on Traffic Data: We validate our approach on large-scale mobile traffic datasets (Milan and Trentino), demonstrating superior generalization, with sLSTM achieving a 30% MAE improvement on unseen data over ConvLSTM.

By combining efficient temporal modeling (sLSTM) with advanced feature fusion, we contribute to the development of more accurate and scalable AI-driven forecasting solutions for next-generation networks.

II. Methodology

Spatiotemporal forecasting is a challenging task that requires capturing both complex spatial correlations and dynamic temporal patterns. Traditional recurrent architectures, such as LSTMs, often face difficulties with long-range dependencies and gradient instability, especially when applied to high-dimensional spatial data. Furthermore, To address these limitations, Zhang and Patras [2] introduced the STN, a dual-path architecture utilizing a ConvLSTM branch alongside a fusion layer to enhance mobile traffic prediction. Our proposed approach incorporate sLSTM into the STN framework, reducing computational overhead and improving predictive performance compared to ConvLSTM.

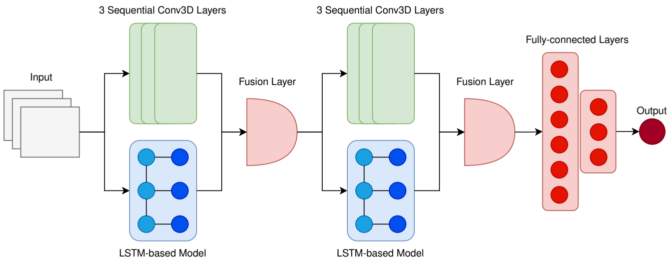

Figure 1: STN Architecture

A. Spatiotemporal Network Architecture

The proposed STN (Figure 1) is designed to model spatial and temporal dependencies. It consists of two parallel branches:

- **Convolutional Branch:** Three sequential Conv3D layers are employed to extract hierarchical spatial features from the grid.
- **Recurrent Branch:** An LSTM-based model—enhanced with a scalar gating mechanism (sLSTM) to stabilize gradients and accelerate convergence—captures the temporal dynamics.

These two pathways are fused in a dedicated fusion layer, integrating spatial-temporal correlations. A fully connected layer processes the fused representation to generate the final output. This design enables the learning of the spatial features hierarchically while preserving temporal dependencies, leveraging both convolutional and recurrent mechanisms. The integration of the sLSTM in the LSTM block helps with stabilizing the gradients and accelerating convergence by using scalar gating, enhancing efficiency without compromising predictive performance. We also assume that an attention-based fusion layer (spatial attention or transformers) can refine feature integration by dynamically weighting spatial-temporal dependencies (capturing long-range correlations). Spatial attention dynamically re-weights spatial dependencies, prioritizing relevant regions during forecasting. Transformer-based fusion, on the other hand, enhances long-range spatial-temporal relationships by capturing higher-order dependencies across time steps. However, due to its higher computational cost, we selectively apply it in controlled experiments to evaluate its trade-offs.

B. Dataset and Preprocessing

The model is evaluated using the Milan mobile traffic dataset [4] from the 2014 Big Data Challenge. This dataset contains mobile phone usage data aggregated over a 100×100 grid at 10-minute intervals. The inherent spatial heterogeneity—characterized by higher activity in the central regions and localized clusters elsewhere—adds complexity to the forecasting task (see Figure 2). In preprocessing, we apply an 11×11 convolutional kernel to capture local spatial dependencies and process sequences of six consecutive time steps to learn temporal patterns. The model then predicts the central value of the kernel for the seventh time step, incorporating both spatial and temporal context for forecasting.

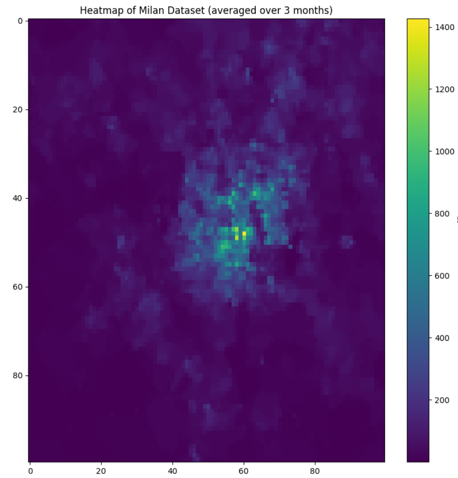

Figure 2: 100x100 Heatmap of Internet Traffic in the Milan Dataset (averaged over 2 months)

C. Evaluation Metrics

The performance of the STN is quantified using the following metrics:

- **Mean Absolute Error (MAE):** Evaluates the average prediction error.
- **Root Mean Square Error (RMSE):** Measures overall deviation, emphasizing larger errors.
- **R-squared (R^2):** Assesses the proportion of variance explained by the model.
- **Structural Similarity Index (SSIM):** Compares the spatial consistency between the predicted and actual 100×100 traffic grids following the reconstruction of the entire spatial grid using model predictions.

Our experimental results benchmark the STN against conventional LSTM models, convolutional LSTMs, and other attention-enhanced architectures, demonstrating its superior balance between accuracy and computational efficiency.

III. Evaluation and Discussion

The models were trained for 40 epochs on 1 million samples extracted from 70% of the data (Milan dataset) using a stride of 6 and tested on 7 million samples from 15% of the data using a stride of 1. The code and model parameters are available at https://github.com/ZineddineBtc/Spatiotemporal-Scalar-LSTM.

We trained the ConvLSTM-based model in three variations: averaging fusion, spatial attention, and transformer-based fusion. The sLSTM-based model was trained in two variations: one with averaging fusion and one with spatial attention, as transformer-based fusion introduces excessive computational overhead when paired with the sLSTM.

Table I shows that sLSTM reduces MAE and RMSE while increasing SSIM and R^2 compared to ConvLSTM. Spatial attention worsens ConvLSTM performance, likely due to

STN Model	MAE	RMSE	R² Score	SSIM
ConvLSTM	7.3917	16.8849	0.9546	0.9853
ConvLSTM + Spatial-Att.	8.7918	19.7825	0.9377	0.9798
ConvLSTM + Transformer	7.2104	15.5359	0.9616	0.9858
sLSTM + Spatial-Att.	5.9738	12.9798	0.9732	0.9901
sLSTM	**5.6704**	**12.3191**	**0.9758**	**0.9912**

Table I: Trained on 1 million points with *stride=6*, tested on 7 million unseen points (Milan dataset).

Step	STN Model	MAE	RMSE	SSIM
1	ConvLSTM	8.9238	33.0079	0.9406
	sLSTM	**6.4451**	**23.4955**	**0.9639**
2	ConvLSTM	11.1397	37.4584	0.9024
	sLSTM	**7.70051**	**24.6218**	**0.9427**
3	ConvLSTM	12.9716	40.4277	0.8781
	sLSTM	**8.8686**	**26.0003**	**0.9373**
4	ConvLSTM	14.4332	44.2043	0.8446
	sLSTM	**9.7918**	**29.6569**	**0.9241**
5	ConvLSTM	16.2411	45.4263	0.8060
	sLSTM	**10.8660**	**32.0609**	**0.9077**
6	ConvLSTM	17.6542	47.3949	0.7651
	sLSTM	**11.7697**	**34.5118**	**0.8917**

Table II: Performance comparison in autoregressive prediction (next-60-minutes prediction in the Milan dataset).

disrupted temporal dependencies. Transformer-based fusion improves RMSE, indicating better feature integration. The sLSTM-based model outperforms all models, achieving the most accurate forecasts while maintaining spatial pattern consistency.

The models were tested on an autoregressive forecasting task, predicting the next 60 minutes iteratively (the following six grids). Table II shows that sLSTM maintains lower errors and higher SSIM across all time steps. ConvLSTM accumulates errors over iterations due to weaker memory retention. RMSE increases while spatial consistency degrades. sLSTM stabilizes temporal dependencies and reduces error propagation. The models trained on the Milan dataset were tested on the Trentino dataset to evaluate inference on categorically unseen datasets. Table III and Table IV show that sLSTM achieves lower MAE and RMSE while maintaining higher SSIM. ConvLSTM performs worse, as it accumulates errors and loses spatial consistency. sLSTM adapts better to new spatial-temporal patterns and maintains stability over multiple inference steps.

IV. Current & Future Work

In this paper, we presented an sLSTM-enhanced STN for 5G mobile traffic forecasting. By integrating a sLSTM branch for efficient temporal modeling and a three-layer Conv3D path

Model	MAE	RMSE	R² Score	SSIM
ConvLSTM	2.6344	7.6370	0.9116	0.9744
ConvLSTM + Spatial-Att.	4.2252	10.2071	0.8421	0.9314
ConvLSTM + Transformer	2.6059	6.8237	0.9294	0.9804
sLSTM + Spatial-Att.	1.8740	5.6004	0.9525	0.9877
sLSTM	**1.6915**	**5.2119**	**0.9588**	**0.9895**

Table III: Performance comparison on an unseen dataset (Trentino).

Step	STN Model	MAE	RMSE	SSIM
1	ConvLSTM	1.6352	7.5786	0.5145
	sLSTM	**1.0811**	**5.5042**	**0.5794**
2	ConvLSTM	1.7053	7.8996	0.4822
	sLSTM	**1.2864**	**6.3942**	**0.5588**
3	ConvLSTM	1.8479	7.5374	0.4476
	sLSTM	**1.4596**	**6.6780**	**0.5416**
4	ConvLSTM	2.3856	10.2821	0.3722
	sLSTM	**1.5980**	**6.3120**	**0.5303**
5	ConvLSTM	2.4865	10.3762	0.3272
	sLSTM	**1.8556**	**7.5945**	**0.5054**
6	ConvLSTM	2.4758	10.5381	0.3041
	sLSTM	**1.9724**	**8.3417**	**0.4954**

Table IV: Performance comparison in autoregressive prediction (next-60-minutes prediction in the Trentino dataset).

for spatial feature extraction—fused via a dedicated fusion layer—our approach stabilized gradient flow compared to ConvLSTM. Empirical evaluations on large-scale 5G datasets demonstrated a 23% MAE reduction over ConvLSTM and a 30% improvement on unseen data, underscoring the robust generalization capabilities of the proposed model.

We further investigated attention-based fusion mechanisms, revealing that transformer-based fusion can enhance long-range feature integration, although its increased complexity may pose scalability challenges in resource-constrained scenarios. In contrast, spatial attention requires careful balancing to avoid disrupting temporal dependencies.

Future work will explore the integration of the Matrix LSTM (mLSTM), as introduced in the same xLSTM framework. We plan to assess its performance when paired with Conv3D to capture spatiotemporal dependencies. Additionally, we aim to cascade mLSTM with sLSTM to form an xLSTM-like structure, leveraging matrix-based transformations to enhance sequence modeling. We will also investigate replacing Conv3D with mLSTM, given its capacity to model spatial relationships while retaining temporal dependencies.

Acknowledgements

Parts of this work has been funded by the Bavarian State Ministry of Education and Culture, Science and Art through the High-Tech Agenda (HTA).

References

[1] 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 *Advances in Neural Information Processing Systems*, 2015.

[2] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," *CoRR*, vol. abs/1712.08083, 2017.

[3] M. Beck, K. Pöppel, M. Spanring, A. Auer, O. Prudnikova, M. Kopp, G. Klambauer, J. Brandstetter, and S. Hochreiter, "xlstm: Extended long short-term memory," 2024.

[4] 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, 2015.