Mobility-Adaptive Digital Twin Modeling for Post-Disaster Network Traffic Prediction

Dong Jia* and Qiang (John) Ye*
*Department of Electrical & Software Engineering, University of Calgary, Calgary, AB, Canada
Emails: dong.jia@ucalgary.ca, qiang.ye@ucalgary.ca

2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall) | 979-8-3315-1778-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/VTC2024-Fall63153.2024.10757774

***Abstract*—In this paper, we propose a mobility-adaptive digital twin (MADT) framework for network traffic prediction in a post-disaster scenario where some affected terrestrial base stations (BSs) are malfunctional, causing unavailable user traffic data under their coverage areas. The MADT framework consists of three core modules: 1) DT data construction, 2) DT modeling for traffic prediction, and 3) DT model calibration. For the data construction, we generate a distribution of user volumes over a considered region, where a clustered modular mobility model (clustered-Mo^3) is tailored to characterize the post-disaster user movements and a spatial-temporal-aware-k-nearest-neighbors (STAK)-based data imputation technique is applied to supplement the missing user traffic data under malfunctional BSs. For traffic prediction, a spatial-temporal graph convolutional network (STGCN) is utilized to establish the DT model under a sequence-to-sequence (seq2seq) forecasting architecture. To improve the traffic prediction accuracy, we further develop a residual calibration (ResCAL) model to estimate and calibrate the traffic prediction errors. The MADT framework establishes a complete DT lifecycle, where the DT model performance is continuously monitored and fed back to trigger the model update if the network traffic pattern changes. Experimental results show that the MADT framework outperforms state-of-the-art spatiotemporal prediction schemes in terms of prediction accuracy and adaptation to user mobility, achieving performance comparable to that of the complete-data-trained STGCN (CD-STGCN).**

I. Introduction

Terrestrial communication infrastructures may experience severe malfunctions after natural disasters (e.g., earthquakes and floods), leading to communication disconnections that impede life-saving rescue operations in disaster-affected regions. Considering the complex network environment and special user mobility patterns, proactive measures for network reconstruction in a timely and cost-effective manner are critical. For example, dispatching unmanned aerial vehicles (UAVs) as temporary communication and computing anchors to cover the affected areas is one such measure. However, implementing proactive network recovery solutions demands an agile, mobility-adaptive, and self-calibrating framework to accurately estimate post-disaster network traffic variations, allowing UAVs to be deployed to proper locations for communication and computing support.

To facilitate research on post-disaster communications, many existing works use customized models for network traffic load characterization. In [1], a stochastic geometry framework is developed to model the distribution of operational terrestrial base stations (BSs) as an inhomogeneous Poisson point process (IPPP) in a post-disaster scenario. A uniform distribution is used to describe the locations of users in [2] and points of interest in [3] after a hazard. The model-based methods balance the trade-off between complexity and accuracy in characterizing network traffic conditions. To improve the accuracy of describing the network-level traffic variations, learning-based traffic prediction methods are recently proposed by exploring the spatial and temporal traffic correlations in real-world datasets, e.g., convolutional long short-term memory (ConvLSTM) [4] and graph-neural-networks (GNN) [5]. Echo state networks are also considered to characterize the temporal dynamics of user movements [6]. Traditional prediction schemes usually perform well when the networks are operational and the user traffic data profiles are available. However, in a post-disaster scenario, some ground BSs can be destroyed or become malfunctional, causing inaccessibility to user traffic data under their coverage areas. Also, user behaviors after a disaster exhibit distinctive features that can deviate from normal patterns, e.g., gathering for rescue and evacuating for survival, and thus cannot be captured by historical data. Therefore, without full access to network-level traffic information, a well-designed user mobility model providing close estimations of post-disaster user movement patterns is important for traffic prediction.

Moreover, a traffic prediction model needs to be adaptive to user mobility patterns in post-disaster scenarios. The exchange of information between the physical network and the traffic prediction model for model updates can be realized using digital twin (DT) technology. The DT framework, featured by creating the high-fidelity digital replica of the physical network with real-time synchronization, can generate a digitalized simulation model for making network management decisions, which significantly reduces communication overhead. Also, through synchronizing with the physical network, the DT model can be periodically refined to adapt to the updated network environment. The DT-enabled network management draws significant attention from the research community. A DT-assisted traffic prediction model proposed in [7] combines deep Q-Learning and generative adversarial networks to handle the highly dynamic and heterogeneous data in transportation systems. An enhanced ConvLSTM model is utilized in [8] to predict the network traffic for load generation and traffic synchronization in DT networks. To promote proactive network management in post-disaster scenarios, DT-assisted network traffic prediction still needs investigation.

Specifically, how to establish a comprehensive DT model for traffic prediction with physical-digital information exchange and performance feedback to achieve a complete DT lifecycle is an important but challenging issue.

In this paper, we propose a mobility-adaptive digital twin (MADT) framework with a closed-loop lifecycle for network traffic prediction in a post-disaster scenario where certain BSs are malfunctional with inaccessible user traffic information. A real-world cellular traffic dataset is considered to initialize the network deployment in an urban region. We employ a clustered modular mobility model (clustered-Mo^3) tailored to describe the user movement patterns after a disaster. Based on the customized mobility model, we establish the MADT framework consisting of three main modules: *DT data construction*, *DT modeling for traffic prediction*, and *DT model calibration*, which are connected through *feedforward* and *feedback* data flowing directions to form a complete DT lifecycle. Specifically, we conduct partial user traffic data refinement to construct the DT dataset, and the spatial-temporal-aware-k-nearest-neighbors (STAK)-based data imputation technique is then applied to supplement the missing user traffic data under malfunctional BSs. For the DT modeling, we conduct spatiotemporal traffic analysis using the sequence-to-sequence (seq2seq) forecasting architecture, where the spatial-temporal graph convolutional network (STGCN) is employed for traffic prediction. The DT model calibration module is developed to estimate and calibrate the prediction errors. Through experiments with a real-world dataset, we demonstrate that the established data-driven and closed-loop DT framework improves traffic prediction accuracy, reduces communication overhead, and adapts well to user mobility through the feedback mechanism.

II. System Model

A. Network Model

We consider a rectangular urban region, denoted by $\mathfrak{R}$, where a homogeneous set of n ($n \in \mathbb{Z}^+$) BSs, denoted by $\mathcal{B} = \{B_1, B_2, \ldots, B_n\}$ are deployed to cover the communication demands from end users. The locations of the BSs are chosen according to a real-world urban traffic dataset which provides information on the BS geographic locations, the number of users associated with each BS, and the request-response records retrieved from the traffic handled by each BS [9]. Each BS is further connected to an edge server, denoted by C_e, through wired backhaul links and the server is mainly responsible for monitoring the BS status and controlling the BS configurations/operations. Based on the locations of the BSs and user traffic information under each BS, we partition the region $\mathfrak{R}$ into non-overlapping Voronoi cells, each representing the coverage area of its corresponding BS [10]. Given the number of users under each BS, users associated with B_i ($i = 1, 2, \ldots, n$) are positioned according to a Poisson Cluster Process (PCP) [11], where user locations are determined to form a cluster centered around the location of B_i, subject to the coverage boundaries and user counts. As shown in Fig. 1(a), an initial network and user deployment are configured according to the real-world BS locations, the Voronoi partitioning, and the PCP model.

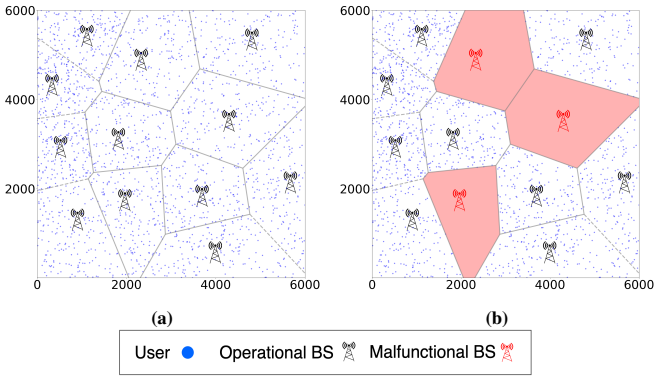

Fig. 1: (a) Pre-disaster network deployment, (b) Post-disaster deployment.

To keep track of user traffic changes, we partition the system time into a sequence of time slots with an identical and fixed length denoted by Δt. For k-th ($k = 0, 1, 2, \ldots$) slot, the starting and ending instants are denoted by t_k and t_{k+1}, respectively. We assume a natural disaster (e.g., an earthquake) happens at time slot 0, affecting the entire region $\mathfrak{R}$ and resulting in possible malfunction on BSs. For B_i, the malfunction probability triggered by the disaster is indicated by φ_i, and we use $\overline{\mathcal{B}} = \{\overline{B}_i, i = 1, 2, ..., n\}$ to indicate the subset of all malfunctional BSs from $\mathcal{B}$. Fig. 1(b) depicts an illustrative scenario where three BSs become malfunctional after the disaster happens. As some BSs are malfunctional without wireless communication capacities, the user traffic information after slot 0 in those affected areas (shaded in red) becomes unavailable at both the malfunctional BSs and the edge server.

B. Mobility Model

To properly characterize the post-disaster user dynamics for network traffic forecasting, we refer to the modular mobility model (Mo^3) [12], which combines multiple human mobility modules to emulate realistic user mobility patterns under various settings. Based on that, we further implement a clustered-Mo^3 model to describe the post-disaster traffic variation as evolving clusters. Due to the intention of user gathering at rescue muster points or urgent evacuation from devastated areas, the affected users/victims are likely to move following cluster patterns, which can experience the process of construction, drift, and decomposition, demonstrating a *nomadic community* mobility pattern. To model this pattern, we tailor the *correlated mobility* module in the clustered-Mo^3 model and incorporate other relevant modules from the original Mo^3 model, including *individual mobility*, *collision avoidance*, and *upper-bounds enforcement*. These modules are configured with scenario-customized parameters to mimic user behaviors within a post-disaster environment. Next, we elaborate on the formulation of the nomadic-community-based correlated mobility module and the user mobility update process.

1) Tailored Correlated Mobility: In the post-disaster scenario, directly applying the correlated mobility module in the Mo^3 model only characterizes the interdependent movements among users, without considering the influence from BSs. Therefore, we propose a refined approach that includes the locations of operational and malfunctional BSs in describing user movements. Any operational BS B_i acts as a cluster center (e.g., a muster point), providing communication connectivities for transmitting critical rescue information to users under its coverage and thus attracting users to gather. We define the corresponding user set $\mathcal{U}_i$ under B_i as a *reference-based cluster* (RBC). On the other hand, any malfunctional BS $\overline{B}_i$ serves as an active cluster center before the disaster. However, the influence of a malfunctional BS on user movement patterns weakens over time after the disaster happens, and the user cluster may exhibit a new cluster centroid. The corresponding user set $\overline{\mathcal{U}}_i$ under $\overline{B}_i$ is defined as a *user-based cluster* (UBC).

To characterize the mobility dynamics of two distinct cluster types, our customized clustered-Mo^3 model incorporates the location information of BSs and only treats the operational BSs as reference points guiding user movement. We define two states of the nomadic community pattern for post-disaster scenarios: *nomadic moving* and *evacuating moving*. A sparse and non-symmetric binding matrix, denoted as BM, is configured to represent these states. An element of $BM(u,k) = 1$ means user u follows the movement of user k, establishing user k as a *mate* of user u. In the nomadic moving state, for each user set $\mathcal{U}_i$, the operational BS B_i is the leader of the community, and any $u \in \mathcal{U}_i$ tends to move towards B_i, which represents the scenario of users assembling for rescue. For example, the binding matrices of one BS (indexed by 0) covering three users (indexed from 1 to 3) in the settings of RBC and UBC are represented by BM_r and BM_u, respectively, which are expressed as

$$BM_r = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad BM_u = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}. \tag{1}$$

From (1), the BSs (row 0 in both matrices) are only correlated with themselves and stay stationary. The users within an RBC (rows 1 to 3 in BM_r) follow both their mates and the BS, whereas users within a UBC (rows 1 to 3 in BM_u) only correlate with their mates. In cases with multiple user clusters, we combine all clusters into a single binding matrix, which is generated according to the user set $\mathcal{U}_i$ and the covering BS B_i. In the evacuating moving state, the binding matrix essentially becomes an identity matrix, which means each user moves according to their own intention and the formed user cluster may drift or disassemble. This state is suitable for illustrating malfunctional BS $\overline{B}_i$, with its users in set $\overline{\mathcal{U}}_i$ venturing for rescue independently due to disconnections. To demonstrate how alternating patterns of nomadic moving and evacuating moving states occupy the time sequence in the proposed mobility model, we further group the partitioned time slot sequence, starting from slot 0, into alternating periods, denoted by $\zeta = \{N_0, E_1, N_2, E_3, \dots\}$ where N_l $(l = 0, 2, 4, \dots)$ represents each nomadic moving period, consisting of N_l slots, and E_p $(p = 1, 3, 5, ...)$ indicates each evacuating moving period, consisting of E_p slots.

2) User Mobility Update Process: As mentioned before, other relevant mobility modules from the Mo^3 model are also incorporated with our tailored correlated mobility module to collectively determine the position updates of any user u in every time slot. The speed vector of user u is denoted as (v_u, θ_u), where v_u is the speed value and θ_u is the moving direction angle of u. The speed vector update process is summarized in the following. More details can be found in [12].

1) *Individual Mobility Module*: Each user's speed vector is fed into this module first, which updates the speed and direction values independently, subject to the maximum speed v_{m}, the maximum speed acceleration α_{m}, and the maximum rotation speed γ_{m}.
2) *Correlated Mobility Module*: For each user u, the speed vector is then modified if the minimum user grouping condition $\rho_{\min}$ is violated and the user's distance from the cluster center is greater than the connected distance D_c. The speed modifications involve rotating the user's direction θ_u towards the cluster center with the rotation speed γ_{m} and adjusting the speed value v_u with the acceleration factor α_{m}.
3) *Collision Avoidance Module*: To prevent potential user collision, this module may further change the speed vector based on the collision trigger distance d^{CA} and the minimum safe distance $d^{CA}_{\min}$.
4) *Upper Bounds Enforcement Module*: This module constrains the user's linear acceleration and angular acceleration within the limits of α_{m} and γ_{m}, respectively.

III. DT-Assisted Traffic Prediction with Model Calibration

A. Proposed MADT Framework

As shown in Fig. 2, we propose the MADT framework with a closed-loop DT lifecycle for predicting the network traffic in a post-disaster scenario. This framework consists of three core modules: *DT data construction*, *DT modeling for traffic prediction*, and *DT model calibration*. These modules integrate two data flows representing the data-driven DT lifecycle: *feedforward* (black solid arrows) and *feedback* (green dash arrows). The feedforward data flow illustrates the process of establishing the main functional components inside each of the DT modules, including partial data refinement, STAK data imputation, user traffic analysis based on the STGCN, and the residual calibration (ResCAL) model. The feedforward data flow aims to forecast network traffic variations after a disaster happens, while the feedback data flow leverages the ResCAL to correct traffic prediction errors from the DT model and continuously monitors the model's performance. If the prediction accuracy drops significantly due to data drift (e.g., a significant change in user mobility patterns), the next DT lifecycle is then triggered to retrain and update the DT model.

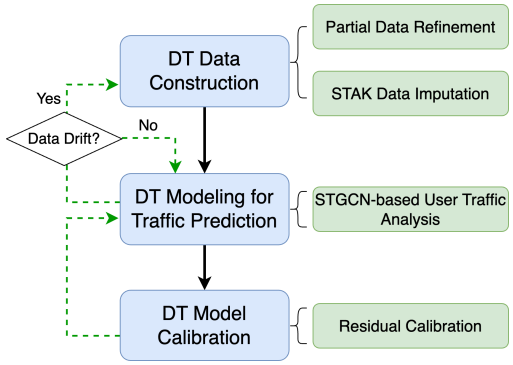

Fig. 2: The proposed framework with a closed-loop DT lifecycle.

The main functional components provided in each of the DT modules are elaborated in the following.

B. DT Data Construction

1) Partial Data Refinement: To construct the DT dataset, we generate a distribution of user volumes (i.e., the number of users) over the considered region $\mathfrak{R}$, based on the user location data obtained through the proposed mobility model. Specifically, we partition $\mathfrak{R}$ into a $W \times L$ square grid, where W and L indicate the total number of square units along the width and length of the region, respectively. We assume each square unit is only covered by one BS. At the k-th time slot, the distribution of user volumes over $\mathfrak{R}$ is represented by a two-dimensional matrix $\mathcal{H}(k)$. The elements of $\mathcal{H}(k)$ are $\{\mathcal{H}_{s_i}(k), s_i \in S_i\}$ and $\{\mathcal{H}_{\overline{s_i}}(k), \overline{s_i} \in \overline{S_i}\}$, indicating the user volumes in respective square units s_i and $\overline{s_i}$ $(i = 1, 2, \ldots, n)$, where S_i and $\overline{S_i}$ denote the sets of units covered by operational B_i and by malfunctional $\overline{B_i}$, respectively.

To reduce the communication cost for data collection, each data collection interval is set to T ($T \in \mathbb{Z}^+$) time slots at the edge server C_e and we assume the data synchronization from each BS to the edge server can be done within one time slot. The partial data collection and refinement process works as follows: For every T time slots, C_e coordinates each operational BS B_i to collect the user location data. B_i then generates the user volume distribution by recording the number of users in each of its covering square units. Finally, C_e combines the distribution data from each online BS B_i into one sample. This process repeats for m ($m \in \mathbb{Z}^+$) collection intervals to create $(m+1)$ consecutive samples. However, for any malfunctional BS $\overline{B_i} \in \overline{\mathcal{B}}$, the user location data under $\overline{B_i}$ are unknown after slot 0, which results in spatially incomplete samples at the edge server. The user volume data in those affected square units under $\overline{B_i}$ are filled with -1. Therefore, the refined user volume distributions over the first m consecutive intervals (i.e., $mT+1$ slots) are denoted by

$$\psi_m = \{\mathcal{H}(0), \mathcal{H}(T), \ldots, \mathcal{H}(mT)\} \tag{2}$$

where the first sample $\mathcal{H}(0)$ at slot 0 is complete and all the other samples have missing data in the areas under the malfunctional BSs.

2) STAK Data Imputation: It can be challenging to accurately predict a complete network-level traffic variation based on incomplete user volume distributions. To address this issue, we propose the STAK data imputation method to enhance the incomplete time-series data ψ_m in (2). For any unit $\overline{s_i} \in \overline{\mathcal{S}_i}$, we select one square unit s_j as its "neighbor" under each operational BS $B_j \in \mathcal{B}$. The selection process is determined using a similarity-based approach. For each unit s_j in the network, we calculate a tuple of two values relative to its serving BS B_j (including malfunctional and operational): the direction angle from B_j to s_j, denoted by ϑ_{s_j}, and the distance from B_j to s_j, denoted by σ_{s_j}. The tuple is then calculated as

$$\begin{cases} \vartheta_{s_j} = \arctan(\frac{y_{s_j} - y_j}{x_{s_j} - x_j}) \\ \sigma_{s_j} = \sqrt{(y_{s_j} - y_j)^2 + (x_{s_j} - x_j)^2} \end{cases} \tag{3}$$

where (x_{s_j}, y_{s_j}) is the coordinates of the center of unit s_j and (x_j, y_j) is the location of B_j. Then, the discrepancy factor $F(\overline{s_i}, s_j)$ for two given units $\overline{s_i}$ and s_j is calculated by

$$F(\overline{s_i}, s_j) = \lambda \frac{|\vartheta_{\overline{s_i}} - \vartheta_{s_j}|}{2\pi} + (1-\lambda)\frac{|\sigma_{\overline{s_i}} - \sigma_{s_j}|}{D} \tag{4}$$

where $|\cdot|$ computes the absolute value of a scalar or the cardinality of a set, D is the diagonal length of the region $\mathfrak{R}$, and $\lambda \in (0, 1)$ is a tunable weighting factor to balance the importance of the angular difference and the distance difference between two square units. The unit s_j under B_j with the smallest $F(\overline{s_i}, s_j)$ is selected as the neighboring unit for $\overline{s_i}$. The set of all neighbors of $\overline{s_i}$ is denoted as $\mathcal{R}_{\overline{s_i}}$. An illustrative example is shown in Fig. 3, where the coverage areas of different BSs are shaded with different colors, and s_1 and s_3 under operational BSs B_1 and B_3 are chosen as the neighboring units, respectively for unit $\overline{s_2}$ under $\overline{B_2}$, based on (3) and (4). Therefore, with the neighboring set $\mathcal{R}_{\overline{s_i}}$, the spatially imputed user volume for unit $\overline{s_i}$ at k-th time slot, denoted by $\mathcal{H}'_{\overline{s_i}}(k)$, is calculated as

$$\mathcal{H}'_{\overline{s_i}}(k) = \frac{1}{|\mathcal{R}_{\overline{s_i}}|} \sum_{s_j \in \mathcal{R}_{\overline{s_i}}} \mathcal{H}_{s_j}(k) \tag{5}$$

where $\mathcal{H}_{s_j}(k)$ is the user volume within the neighbor unit s_j at the k-th time slot. Once all missing user volumes are imputed, a spatially complete user volume distribution $\mathcal{H}'(k)$ over region $\mathfrak{R}$, is obtained for the k-th ($k > 0$) time slot. On the temporal axis, the STAK data imputation starts from the initial user volume distribution, and calculates subsequent distributions sequentially, thus producing spatially complete and temporally continuous data for user traffic prediction.

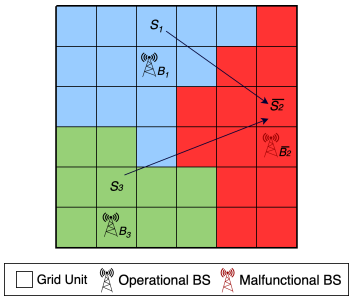

Fig. 3: A grid partition on the network region for DT data construction.

C. DT Modeling for Network Traffic Prediction

The DT prediction module leverages the constructed DT data to forecast the future distributions of user volumes over the entire region $\mathfrak{R}$. We formulate the user traffic prediction problem using a spatiotemporal sequence-to-sequence (seq2seq) forecasting architecture. This architecture takes a sequence of user volume distribution samples, denoted by $\mathbf{X} \in \mathbb{Z}^{P\times W\times L}$, as input and predicts the future sequence of the user volume distributions, denoted by $\hat{\mathbf{Y}} \in \mathbb{Z}^{F\times W\times L}$, where $\mathbb{Z}$ indicates the integer number data type for each dimension, P represents the input sequence length, F is the output sequence length. To incorporate the location information of BSs into the prediction architecture, we utilize an STGCN-based approach to establish the DT model.

1) STGCN-based Traffic Prediction: As shown in the upper half of Fig. 4, the STGCN model [13] comprises multiple spatiotemporal blocks (ST blocks) and each block utilizes a three-layer structure, where one spatial graph convolutional layer (Spatial Graph-Conv) is wrapped by two temporal gated convolutional layers (Temporal Gated-Conv). Apart from the sequence of user volume distribution samples, a graph adjacency matrix, denoted by G representing the spatial dependencies of square units, is taken as the input of each ST block. Multiple ST blocks are arranged sequentially, providing latent space mappings for the output block, which consists of a Temporal Gated-Conv and several fully connected layers to generate the future user volume distribution sequence. All of these blocks incorporate normalization and dropout layers for post-processing.

In Spatial Graph-Conv, each square unit within $\mathfrak{R}$ is treated as an observation point and the element $G(s_i, s_j)$ in the adjacency matrix is calculated based on the distance between unit s_i and s_j. However, this method produces a homogeneous similarity measurement among all square units, which only captures spatial relationships without considering the user distribution patterns and the influence of BSs. To address this, we use the locations of BSs as centers to compute the similarity among the square units using the formula introduced in [13]. This approach results in a sparse and symmetric matrix G, where only units near BSs have non-zero values, effectively capturing the impact of BS proximity on unit similarity. To generate user volume sequences, we use a sliding window (with the size of $P + F$) to partition the DT dataset ψ_m. Starting from the first sample $\mathcal{H}(0)$, the data samples falling within the window are considered as one data sequence, which is further split into the input and target parts for supervised learning. Then, the window slides forward with a fixed stride of I samples to create a new sequence. This process repeats until the window reaches the last element in ψ_m, producing $\frac{m-(P+F)}{I} + 1$ sequences in total. In our approach, we set $F < P$ to provide a longer length of historical input data than that of the output predictions. For compensation, a sliding window inference process is employed. During each inference step, the model predicts a data sequence of length F as an intermediate output. The input $\mathbf{X}$ then slides forward by F samples to form a new input sequence for the next prediction. This process iterates until the total length of the predicted output sequences reaches the original input length P. These intermediate outputs are then concatenated on the temporal dimension to form the final predicted output.

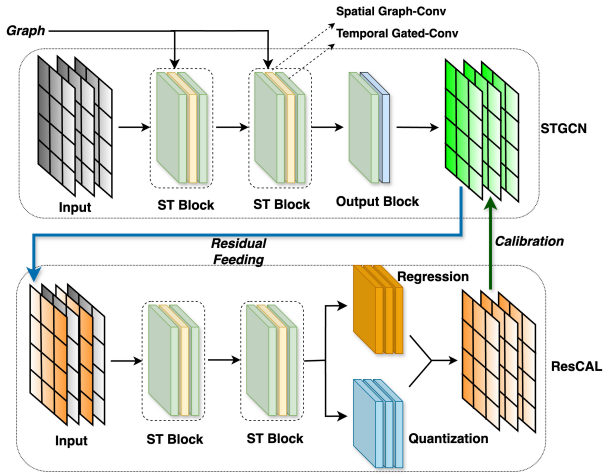

Fig. 4: Architectures and correlations of the STGCN and ResCAL models.

D. DT Model Calibration

To improve the traffic prediction accuracy of the established DT model, we further develop a DT calibration module based on ResCAL to estimate and calibrate the traffic prediction errors [14]. We employ a two-stage training approach: The DT model is trained in the first stage, and the second stage is used to train the ResCAL model with the residuals obtained from the prediction outputs. Similar to the network traffic prediction, the residual estimation and calibration problem is formulated using the spatiotemporal seq2seq forecasting architecture, where the past user volume distributions and the prediction residuals of length P are taken as input and the output is a predicted residual sequence of length F.

1) Residual Calibration Model: In the ResCAL model, ST blocks are also employed to extract the spatiotemporal correlations to a hidden representation which is then fed to a regression branch and a quantization branch separately, as shown in the lower half of Fig. 4. The user volume sequence and the residual sequence are concatenated on the temporal dimension as the input data, and the same graph adjacency matrix is also fed to the ST blocks, similar as in the STGCN model. The regression branch forecasts the continuous values of the DT model's prediction errors, while the quantization branch provides interpretation of the failures from the DT model. The output of the regression branch is directly used, while the output of the quantization branch is passed to a straight-through Gumbel estimator and is multiplied by a learnable embedding vector. Finally, the summation of the two branch outputs is explored by the output layer which produces the final estimated residuals. Both branches and the output layer are constructed with point-wise convolutional layers (i.e., a convolution layer with a kernel size of 1). More details on the model structure can be found in [14]. The sliding window technique is also utilized in the ResCAL model to partition the training sequences and extend the inference sequences.

Main Parameters for Mobility Simulation	
Maximum speed value (v_{m})	2 m/s
Maximum speed acceleration (α_{m})	1 m/s^2
Maximum rotation speed (γ_{m})	$\pi/2$ rad/s
Connected distance (D_c)	300 m
Minimum user grouping density ($\rho_{\min}$)	0.8
Collision trigger distance (d^{CA})	3 m
Collision minimum safe distance ($d^{CA}_{\min}$)	1 m
Nomadic moving period (N_l)	80 s
Evacuating moving period (E_p)	800 s
Main Parameters for MADT Framework	
Data sample collection interval (T)	5 s
Square unit size	100×100 m^2
The weighting factor in STAK (λ)	0.5
Input and output sequence length (P, F)	(12, 1)
Number of ST blocks	2
Channels for one ST block	[64, 16, 64]
Spatial and temporal kernel size	3
Output channels after ST blocks in ResCAL	128
Batch size	16
Training epochs	60
Learning rate	0.001

TABLE I: Important experiment parameters

IV. Experimental Results

A. Experiment Settings

1) Simulated Network Environment: We consider a real-world urban region with the size of 6000×6000 m^2. There are 12 BSs and 2076 users deployed within the region according to the real-world dataset[1] and the network model specified in Section II-A. For each BS B_i, we set the malfunction probability φ_i to 0.25, resulting in 9 operational BSs and 3 malfunctional BSs. Our tailored clustered-Mo^3 mobility model is employed to simulate the user movement patterns after a disaster. For the considered time-slotted system, we set each time slot, Δt, to 1 second and the total simulation time to 14400 seconds. Other important parameters used in simulating user mobility are detailed in the upper half of Table I.

2) DT Framework: The parameters used in MADT are listed in the lower half of Table I. Given the sizes of the region $\mathfrak{R}$ and square units, each user volume distribution sample is refined and represented by a 60×60 matrix and is then normalized by a maximum user count of 100. The DT dataset is further split into a training set of 2660 samples and a test set of 220 samples. The MADT framework is built on available user traffic data under operational BSs to reflect a post-disaster scenario, while the test traffic data are assumed complete over the entire region for evaluation. The learning-related hyperparameters apply to both the STGCN and ResCAL models, which are trained to minimize the mean absolute error (MAE) using the AdamW optimizer. Notably, the Spatial Graph-Conv inside each ST block is implemented with a Chebyshev polynomial approximation to ensure accurate predictions [13].

B. Performance Evaluation

We first evaluate the accuracy of MADT in predicting the spatial distribution of the user volumes. The ground truth data and the predicted user volume distribution over the entire region are shown using heatmaps in Fig. 5, where we can see that the MADT prediction output accurately captures the spatial distribution of user volumes. To assess the accuracy of using STAK for data imputation under malfunctional BSs, we show in Fig. 6 a performance comparison between the MADT framework and the complete-data-trained STGCN (CD-STGCN) model, which assumes that the user traffic information under the malfunctional BSs are also available. We can see that the predicted user volumes from both models are close to each other over time in areas covered by an operational BS and by a malfunctional BS, which are also well aligned with the ground truth data.

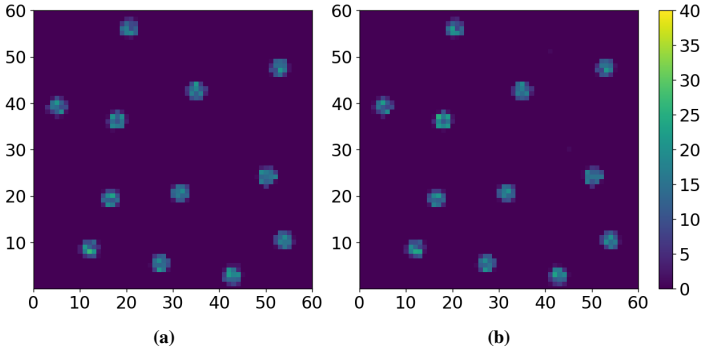

Fig. 5: User volume distribution: (a) Ground truth, (b) MADT prediction.

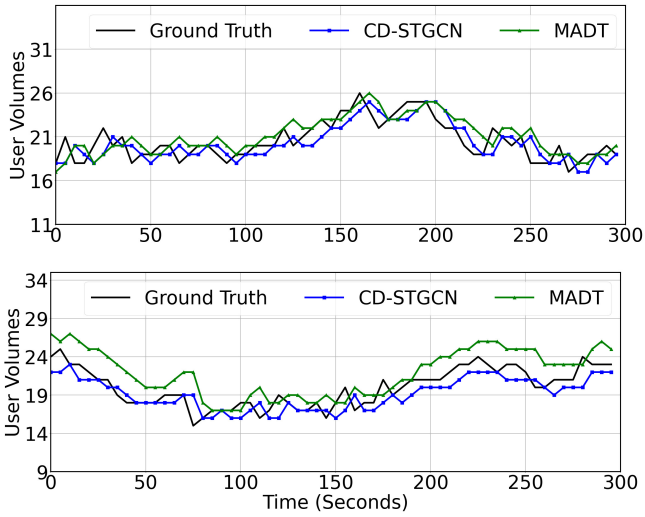

Fig. 6: Comparison of user volume prediction in the areas covered by one operational BS (upper) and one malfunctional BS (lower).

Lastly, we compare the MADT with one state-of-the-art spatiotemporal traffic forecasting model, i.e., the ConvLSTM model [4] and with the MADT without residual calibration. Table II shows a prediction performance comparison by evaluating the prediction errors using the following four metrics: 1) **MAE**, the average difference between predicted and observed values without considering the spatial correlations; 2) **Root mean squared error (RMSE)**, the quadratic mean of the differences between the predicted and observed values; 3) **Mean absolute scaled error (MASE)**, a scaled MAE for the comparison among different models or scales; 4) **Structural similarity index measure (SSIM)**, the spatial similarity between predicted and observed values, with higher values indicating more similar structures. From Table II, we observe

[1]https://github.com/caesar0301/city-cellular-traffic-map

Method	MAE	RMSE	MASE	SSIM
ConvLSTM	0.0916	0.4816	0.2385	0.9956
MADT without ResCAL	0.0671	0.3803	0.1500	0.9969
MADT	**0.0644**	**0.3735**	**0.1424**	**0.9971**
CD-STGCN	0.0603	0.3263	0.1396	0.9972

TABLE II: Performance comparison among different prediction approaches.

that the MADT achieves the lowest values of MAE, RMSE, and MASE among the prediction schemes built on actual collected data, indicating both the average prediction errors and the average error deviations are the smallest. The SSIM of MADT is among the highest, showing a high spatial similarity between the predicted and real user volume distributions. The CD-STGCN model, trained on assumed complete data, serves as a benchmark. Our MADT achieves performance comparable to CD-STGCN on all metrics, particularly in terms of the SSIM indicator. Fig. 7 shows a comparison of predicted user volume distributions over time among different schemes. It

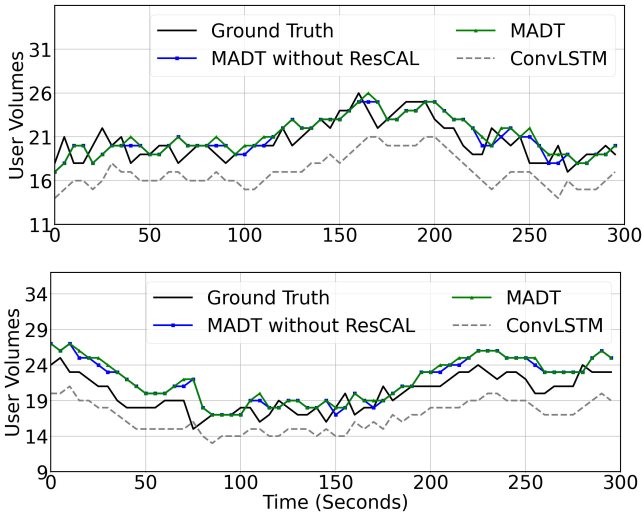

Fig. 7: Comparison of schemes for user volume prediction in areas covered by one operational BS (upper) and one malfunctioning BS (lower).

can be seen that with model calibration, MADT captures the overall variations of user traffic better than the one without residual calibration. By explicitly incorporating the spatial correlation between user volumes and square units, as well as the influence of BSs, MADT generates more accurate prediction results than the ConvLSTM. The advantages are more notable in the areas covered by operational BSs. The reasons are 1) the STAK imputed data under malfunctional BSs have some accuracy gap with the real data and 2) the input graph structure of STGCN capturing BS locations is static, whereas the underlying graph structures of user clusters under malfunctional BSs shift a bit from the BS locations as time evolves. The observations from Fig. 7 indicate that the MADT performance is well adapted to traffic variations both spatially and temporally, compared with the state-of-the-arts.

V. Conclusion

In this paper, we have proposed an MADT framework for network traffic prediction in a post-disaster scenario. The MADT framework establishes a complete DT lifecycle which consists of DT data construction, DT modeling for traffic prediction, and DT model calibration. For data construction, we refine from a real-world user traffic dataset to generate spatiotemporal user volume distributions over a considered network region, by tailoring the clustered-Mo^3 to describe post-disaster user mobility patterns and by using the STAK to supplement the missing data under the malfunctional BSs. Based on the DT dataset, we then develop the DT-assisted traffic prediction module based on the STGCN prediction architecture where the BS locations are incorporated into the prediction module. To further improve the prediction accuracy, the ResCAL model is added to calibrate the traffic prediction errors. The prediction performance is consistently monitored and fed back to trigger the DT model update. Experimental results demonstrate the superiority of our DT model in prediction accuracy and adaptation to user mobility. For future work, we will investigate how the MADT framework can be integrated with the design of proactive UAV deployment for post-disaster network recovery.

References

[1] M. Matracia, M. A. Kishk, and M.-S. Alouini, "UAV-aided post-disaster cellular networks: A novel stochastic geometry approach," *IEEE Trans. Veh. Technol.*, vol. 72, no. 7, pp. 9406–9418, Jul. 2023.

[2] Y. Zhu and S. Wang, "Joint deployment and trajectory planning of multiple uavs for emergency communications," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Dec. 2023, pp. 1854–1859.

[3] Q. Guo, J. Peng, W. Xu *et al.*, "Minimizing the longest tour time among a fleet of uavs for disaster area surveillance," *IEEE Trans. Mob. Comput.*, vol. 21, no. 7, pp. 2451–2465, Jul. 2022.

[4] W. Jiang, "Internet traffic matrix prediction with convolutional LSTM neural network," *Internet Technol. Lett.*, vol. 5, no. 2, p. e322, 2022.

[5] Z. Wang, J. Hu, G. Min *et al.*, "Spatial-temporal cellular traffic prediction for 5G and beyond: A graph neural networks-based approach," *IEEE Trans. Ind. Inform.*, vol. 19, no. 4, pp. 5722–5731, Apr. 2023.

[6] H. Peng, A.-H. Tsai, L.-C. Wang *et al.*, "LEOPARD: Parallel optimal deep echo state network prediction improves service coverage for UAV-assisted outdoor hotspots," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 1, pp. 282–295, Mar. 2022.

[7] L. Nie, X. Wang, Q. Zhao *et al.*, "Digital twin for transportation big data: A reinforcement learning-based network traffic prediction approach," *IEEE Trans. Intell. Transport. Syst.*, vol. 25, no. 1, pp. 896–906, Jan. 2024.

[8] J. Lai, Z. Chen, J. Zhu *et al.*, "Deep learning based traffic prediction method for digital twin network," *Cogn. Comput.*, vol. 15, no. 5, pp. 1748–1766, Sep. 2023.

[9] X. Chen, Y. Jin, S. Qiang *et al.*, "Analyzing and modeling spatio-temporal dependence of cellular traffic at city scale," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 3585–3591.

[10] K. Feng and M. Haenggi, "Joint spatial-propagation modeling of cellular networks based on the directional radii of poisson voronoi cells," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 5, pp. 3240–3253, May 2021.

[11] C. Saha, M. Afshang, and H. S. Dhillon, "Poisson cluster process: Bridging the gap between PPP and 3GPP HetNet models," in *Proc. Inf. Theory Appl. Workshop*, Feb. 2017, pp. 1–9.

[12] L. De Nardis and M.-G. D. Benedetto, "Mo3: A modular mobility model for future generation mobile wireless networks," *IEEE Access*, vol. 10, pp. 34 085–34 115, 2022.

[13] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Jt. Conf. Artif. Intell. (IJCAI)*, Jul. 2018, pp. 3634–3640.

[14] D. Kim, Y. Cho, D. Kim *et al.*, "Residual correction in real-time traffic forecasting," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manag. (CIKM'22)*, New York, NY, USA, Oct. 2022, pp. 962–971.