Spatial-Temporal Graph Representation Learning for Tactical Networks Future State Prediction

Junhua Liu*†, Justin Albrethsen*, Lincoln Goh†, David Yau*, Kwan Hui Lim*

*Singapore University of Technology and Design

{junhua_liu, justin_albrethsen, david_yau, kwanhui_lim}@sutd.edu.sg

†Forth AI,

{j, lincoln}@forth.ai

‡Singapore Polytechnic

Abstract—Resource allocation in tactical ad-hoc networks presents unique challenges due to their dynamic and multi-hop nature. Accurate prediction of future network connectivity is essential for effective resource allocation in such environments. In this paper, we introduce the Spatial-Temporal Graph Encoder-Decoder (STGED) framework for Tactical Communication Networks that leverages both spatial and temporal features of network states to learn latent tactical behaviors effectively. STGED hierarchically utilizes graph-based attention mechanism to spatially encode a series of communication network states, leverages a recurrent neural network to temporally encode the evolution of states, and a fully-connected feed-forward network to decode the connectivity in the future state. Through extensive experiments, we demonstrate that STGED consistently outperforms baseline models by large margins across different time-steps input, achieving an accuracy of up to 99.2% for the future state prediction task of tactical communication networks.

Index Terms—Spatial-Temporal Graph Neural Networks, Representation Learning, Tactical Communication Networks, Future State Prediction

I. INTRODUCTION

Modern militaries require reliable communications on the tactical edge to support dynamic operations. Tactical communication networks (TCNs) extend 4G, 5G, and SATCOM communication capabilities to units without access to traditional network infrastructure. To this end, TCNs are typically formed as self-organizing mobile ad-hoc networks (MANETs) using high frequency (HF), very high frequency (VHF), and ultra high frequency (UHF) radio technologies [1]. In TCNs it is critical to distribute information to support accurate and updated situational awareness. In these circumstances poor Quality of Service (QoS) such as dropped packets or high delays can lead to catastrophic military outcomes.

QoS is critical to TCNs and is also challenging to provide. One example of a QoS technique is RSVP which allows users to reserve networking resources for all nodes in an application stream's path [2]. This relies on established routes and ample networking resources to guarantee QoS. However, TCNs often have low data-rates and routes consisting of multiple wireless hops with varying signal strengths [3]. Furthermore, nodes are mobile across diverse terrain and may frequently break routes without warning, which makes the topology highly dynamic. Even standard protocols like TCP can become ineffective when used in this kind of dynamic networks [4]. TCNs are highly dynamic and routes often involve multiple hops, as such it is difficult to guarantee resources to every node involved in sending or routing critical applications. Forecasting future network connectivity in TCNs is a necessary step to enable resource reservation and QoS guarantees for priority network traffic [5].

TCNs introduce unique challenges to traditional networking QoS solutions. To study these problems, the North Atlantic Treaty Organization (NATO) created the Anglova scenario [6]. The Anglova scenario is designed to represent typical communication challenges in real TCNs and provide a standard experimentation environment for research [1] [7] [3]. The Anglova scenario includes sensor networks, naval task force, tank battalions, mechanized infantry, aerostat, UAVs, and medivac nodes. The scenario is organized into three vignettes to showcase different operational requirements.

Within this scenario we look to predict future network connectivity as a precursor for future QoS work. Predicting future network connectivity can be formulated as a kind of link prediction problem, which has many successful solutions among a diverse group of domains including airline routes [8], social media [9], criminal networks [10], and e-commerce [11]. Link prediction has also been successful with certain MANETs using graph networks which can capture both spatial and temporal relationships that are present in dynamic networks [12]. Existing work does attempt to predict future link quality in tactical networks, but their model assumes a simplified free space radio propagation model and does not account for any terrain effects [13]. As part of the Anglova scenario, they provide radio path loss data based on a propagation model that incorporates complex terrain, making link prediction more complicated and realistic.

This work aims to predict future connectivity in the realistic Anglova scenario. In summary, the main contributions include:

- **Datasets for TCNs:** Introduction of two benchmark datasets, Communication Network with Tactical Movement (CNTM) and Communication Network with Casual Movement (CNCM), based on the Anglova scenario to stimulate future TCN research [1].

[1]available at: https://github.com/junhua/STGED

- **STGED Framework:** Introduction of STGED framework that integrates Graph Neural Networks (GNNs) with Recurrent Neural Networks (RNNs) to effectively learn a spatial-temporal representation of dynamic tactical networks. To the best of our knowledge, we are the first to propose a graph encoder-decoder framework for learning representations of tactical communication networks.
- **Experimental Validation:** Through comprehensive experiments, we demonstrate the superior performance of STGED, achieving 99.2% accuracy in predicting future network connectivity and outperforming baselines of recent works.

II. RELATED WORK

A. Tactical Communication Networks

A considerable body of research has applied the Anglova scenario to study TCNs. Lopatka et al. [14] focused on spectrum optimization to mitigate congestion, while Suri et al. [15] improved Blue Force Data dissemination through optimized link state routing (OLSR) and simplified multicast forwarding. Addressing the importance of communication reliability, Schutz et al. [16] proposed a multi-path forward erasure correction method based on network coding. In contrast, Rechenberg et al. [13] utilized recurrent neural networks (RNNs) for link prediction but did not consider tactical mobility models or terrain effects, which are critical in TCNs.

B. Link Prediction in MANETs

Link prediction has been widely studied across various domains, including MANETs, which share similarities with TCNs. Preetha et al. [17] used fuzzy decision processes and belief propagation to predict link duration. Patel et al. [18] employed statistical techniques based on signal strength for the same purpose. Furthermore, Shao et al. [12] combined graph convolutional networks (GCN) with gated recurrent units (GRU) to capture spatial-temporal features in dynamic networks.

C. GNNs in Networking

GNNs have shown promise in various networking applications. Maret et al. [19] and Swaminathan et al. [20] applied GNNs to optimize routing in the Anglova scenario and demonstrated generalizability across different network topologies. Sun et al. [21] optimized virtual network function placement using a GNN with reinforcement learning (RL). Applications extend to Internet of Things (IoT) networks, where GNNs have been used for optimal resource allocation [22] and traffic prediction [23].

D. Spatial-Temporal Graph Neural Networks (STGNN)

Special-temporal models are widely used to model sequential applications in the natural language processing domain, such as itinerary planning [24], social media mining [25]–[27], geo-tagging [28], [29], and other applications [30]–[32]. In networking, STGNNs are relevant for capturing data correlation in both time and space. For example, Chen et al. [33] proposed an STGNN combining GCN and long short-term memory (LSTM) unit for cellular network traffic prediction. Wang et al. [34] outperformed traditional GNN and GRU models in predicting cellular traffic. In the context of road traffic forecasting, Lan et al. [35] introduced a dynamic STGNN that significantly outperformed state-of-the-art methods. Li et al. [36] proposed a fusion operation on multiple spatial and temporal graphs for synchronized learning of hidden spatial-temporal correlations. Li et al. [37] proposed a graph encoder-decoder approach to conduct fault diagnosis in industrial processes.

E. Summary

The aforementioned works demonstrate significant advancements in TCNs, link prediction in MANETs, and the application of GNNs in various networking contexts. However, there is a gap in the literature concerning the effective incorporation of spatial-temporal modeling in TCNs for enhanced performance and reliability. Motivated by the exceptional performance of STGNNs in other domains, this paper aims to bridge this gap by applying advanced spatial-temporal modeling techniques to TCNs.

III. METHODOLOGY

A. Preliminary

We denote the Tactical Communication Network (TCN) as a temporal graph $G^{(t)} = (V^{(t)}, E^{(t)}, X_V^{(t)}, X_E^{(t)})$, where $V^{(t)} \in \mathbb{R}^N$ represents a finite set of N nodes (i.e. communication devices) within the TCN; $E^{(t)} \in \mathbb{R}^M$ is a set of M edges that represent valid communications between pairs of nodes; $X_V^{(t)} \in \mathbb{R}^{C_X}$ denotes the set of node features (i.e. velocity) at time t; and $X_E^{(t)} \in \mathbb{R}^{C_W}$ is the TCN's connectivity features (i.e. distance, path loss, prop-delay and timestamp) at time t.

This work aims to predict the connectivity status E_{t+1} at time $t+1$, i.e., whether a link exists for a given pair of node, based on a window of past observations of the network states, i.e., $G_{(t-w)...t}$. Our goal is to train a STGED model to predict the network connectivity, i.e. links among given pairs of nodes, in the next state, as follows:

$$
\begin{aligned}
E_{t+1} &= STGED[G^{(t-w):t}] \\
&= STGED[V^{(t-w):t}, E^{(t-w):t}, X^{(t-w):t}, W^{(t-w):t}]
\end{aligned}
\tag{1}
$$

B. Spatial-Temporal Graph Encoder-Decoder Framework

We present the framework of Spatial-Temporal Graph Encoder-Decoder (STGED) for TCNs in Figure 1. The Encoder-Decoder architecture is widely used in the Computer Vision (CV) and Natural Language Processing (NLP) communities for sequence classification tasks, such as machine translation and image captioning. However, as the features of TCN data are physical measures and do not contain explicit semantics like that in CV and NLP, learning a feature representation for the latent tactical behavior becomes more challenging. Therefore, we propose the Spatial-Temporal
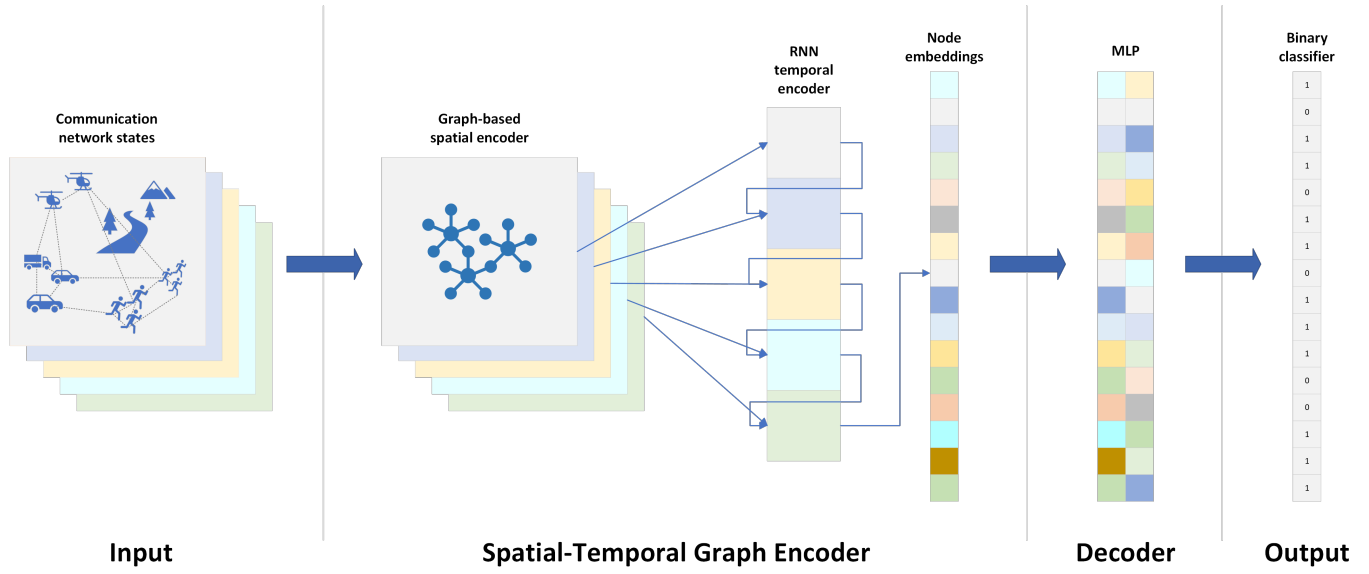
Fig. 1: A Spatial-Temporal Graph Encoder-Decoder (STGED) framework for modelling Tactical Communication Networks (TCN). A series of TCN states pass through a graph-based neural network to learn the spatial features for each state. Subsequently, the states form a time-series input and passes through a recurrent neural network (RNN) to learn the temporal features, i.e. the tactical movement patterns. The last hidden layer of the RNN acts as the embeddings of nodes in the TCN. The decoder takes pairs of node embeddings as input and predicts their connectivity at the next time step.

Graph Encoder-Decoder (STGED) framework to address this challenge. Intuitively, we first use a graph-based convolution network to learn the spatial representation of each network state. Subsequently, these state representations are passed as time-series inputs to a recurrent neural network to capture latent temporal features emerging from the evolution of the input states.

Formally, the Encoder comprises two main components: a graph-based encoder and an recurrent temporal encoder. These components work together to learn a network embedding $Z_{(t-w):t}$ based on the node and edge features, as follows:

$$Z^{(t-w):t} = TE[SE[G^{(t-w):t}]] \qquad (2)$$

The Decoder then takes in the spatial-temporal network embedding and predict the network connectivity at $t + 1$:

$$E^{(t+1)} = DE[Z^{(t-w):t}] \qquad (3)$$

C. Spatial-Temporal Graph Encoder

In TCN, both the communicating devices and their connectivity display spatial relationship in the network, such as the relative velocities and distance among connected nodes. We adopt a Graph Transformer Convolutional (GTC) unit [38] that uses multi-head attention mechanism to compute the latent relationship among all the nodes. Concretely, the GTC unit computes the node representation x_i' of ith node where $i < N$ by propagating information from the ith node and all other jth nodes $j < N$, $j \neq i$, thereby learning the global latent temporal features:

$$x_i' = \mathbf{W}_1 \mathbf{x}_i + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j} \left(\mathbf{W}_2 \mathbf{x}_j + \mathbf{W}_3 \mathbf{e}_{ij} \right) \qquad (4)$$

and attention coefficients $\alpha_{i,j}$ as:

$$\alpha_{i,j} = \mathrm{softmax}\left(\frac{(\mathbf{W}_4 \mathbf{x}_i)(\mathbf{W}_5 \mathbf{x}_j + \mathbf{W}_6 \mathbf{e}_{ij})}{\sqrt{d}} \right) \qquad (5)$$

where $x_i, x_j \in X_E$, edge features $e_{ij} \in X_V$, $\sqrt{d}$ as a scaling factor for d dimensions, and $\mathbf{W}_{1:6}$ are trainable parameters.

After that, a series of the GTC units from time $t-w$ to t will then be passed into a LSTM to learn the *temporal* behaviours from the evolution of network states:

$$\mathbf{h}_t, \mathbf{c}_t = \mathrm{LSTM}(\mathbf{W}, \mathbf{b}, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}, \mathbf{x}_t) \qquad (6)$$

where $\mathbf{W} = \{\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_o, \mathbf{W}_g\}$ are trainable weight metrics, $\mathbf{b} = \{\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_o, \mathbf{b}_g\}$ are trainable bias factors, for the LSTM's forget, input, output, and candidate cell state, respectively. The LSTM calculates the new hidden state h_t and cell state c_t previous hidden state $\mathbf{h}_{t-1}$ by previous cell state $\mathbf{c}_{t-1}$, $\mathbf{x}_t'$.

Finally, we take the last hidden state of the LSTM as the TCN's state encoder from time $t - w$ to t, i.e.:

$$Z^{(t-w):t} = \mathbf{h}_t \qquad (7)$$

We set $Z^{(t-w):t} \in \mathbb{R}^{N \times H}$, where N is the number of nodes and H is the node embedding size, which is a tuneable hyperparameter. This implicitly instructs the STGE to learn effective embeddings of each node. As a result, we could simply pass embeddings of two nodes into the decoder to predict their connectivity at a future state.

D. Multi-Layer Perceptron Decoder

The decoder module is designed to predict the connectivity between pairs of nodes in the graph. Specifically, it takes as input the node embeddings generated by the LSTM encoder and outputs a connectivity score for each pair of nodes. Concretely, the decoder can be formulated as follows:

Let $\mathbf{emb}^{(i)}$ and $\mathbf{emb}^{(j)}$ be the embeddings for nodes i and j respectively, where $\mathbf{emb}^{(i)}, \mathbf{emb}^{(j)} \in \mathbb{R}^H$ and H is the dimension of node embeddings.

These vectors are then passed through the MLP decoder to obtain the connectivity score s_{ij}:

$$s_{ij} = \mathrm{MLP}(\mathbf{W}, \mathbf{b}, \mathbf{Em}) \tag{8}$$

where $\mathbf{Em} = concat(\mathbf{emb}^{(i)}, \mathbf{emb}^{(j)})$.

Here, MLP is a function representing the Multi-Layer Perceptron, parameterized by weight matrices $\mathbf{W}$ and bias vectors $\mathbf{b}$. The function takes the concatenated node embeddings $\mathbf{z}_{ij}$ as input and outputs a scalar s_{ij} that represents the predicted connectivity between nodes i and j.

At training time, the output s_{ij} is kept as the probability of the connectivity that will then be passed into a loss function. At testing time, the score is then thresholded to make a binary decision about whether nodes i and j are connected or not. Specifically, if s_{ij} is greater than a predefined threshold τ the nodes are considered to be connected.

$$\mathrm{Connectivity}_{ij} = \begin{cases} 1 & \text{if } s_{ij} > \tau \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

To allow the decoder module to effectively map the high-dimensional embeddings to a connectivity prediction, we perform next-state one-hop connectivity prediction task with a Binary Cross Entropy (BCE) loss function with $\tau = 0.5$[2].

IV. EXPERIMENTS

A. Datasets

For our TCN experiments we generated datasets using EXata [39], a high fidelity network emulation toolset. EXata uses a software virtual network to represent networks across various protocol levels and radio types, and can even interoperate with real devices to provide hardware-in-the-loop capabilities. We used EXata to model a subset of the Anglova scenario. Table I summarizes the statistics of both datasets.

1) Communication Network with Tactical Movements (CNTM): For our first dataset, named Communication Network with Tactical Movements (CNTM), we used the subset of the Anglova scenario including a single tank company using UHF radios during vignette 2, which is the troop deployment stage. For this stage we used node position information and a radio path loss matrix provided on the Anglova website [6]. This data has a granularity of one second and a duration of 7200 seconds, or 2 hours. In EXata we configured the nodes to follow the provided node positions and use the provided path

---

[2]Empirically, the convergence rate and performance of the model are not sensitive to the choice of the threshold.

	CNTM	CNCM
#States	3998	3599
Avg. Nodes	24.0	24.0
Avg. Edges	685.20	1280.55
Node features	VelocityX, VelocityY	
Edge features	Distance, Path loss, PropDelay, Timestamp	

TABLE I: Statistics of CNTM and CNCM datasets. Note that the Avg. Edges are more than a fully-connected graph with 24 nodes in one-second interval. Within each time step, there could exist none, one or many edges between any two nodes.

loss. We also configured the UHF radios, the time division multiple access (TDMA) protocol, and OLSRv2 routing protocol per the Anglova definitions [3]. EXata can be configured to log events that occur at the physical layer, MAC layer, network layer, and application layer. In addition to the events, EXata also can regularly save status tables at user defined granularity. These events and status tables across the 2 hour simulation make up the dataset and are saved in SQLite databases for further processing. During the experiment, the majority of the traffic is generated by the OLSRv2 routing messages, HELLO messages that nodes broadcast every 2 seconds, and Topology Control (TC) messages every 8 seconds. This is enough to keep connectivity states updated at a minimum of every 2 seconds. In the future, additional tactical applications such as Blue Force Tracking will be added.

2) Communication Network with Casual Movements (CNCM): The second dataset, named Communication Network with Casual Movements (CNCM), is setup identically to the CNTM dataset except for node positions, mobility, and path loss. The mobility uses a random waypoint model provided in EXata, with a maximum speed of 10 meters per second. Because this does not use the Anglova positions, we cannot use the Anglova path loss values. Instead we use the two-ray ground reflection radio propagation model provided by EXata. This model is similar to a free space propagation model, with the addition of interference reflected from the ground.

3) Visualisation: As shown in Figure 2, the random nature of the CNCM dataset and the objective-oriented mobility of the CNTM dataset. Note that the arrows between nodes represent messages being sent successfully. Nodes that are not connected cannot successfully send messages, but not all nodes that are connected will be sending messages at any given snapshot. Also note that the scale is different, as the CNTM travels a greater distance, but generally is grouped together. If the same scale was used for the CNCM dataset, there would be much fewer connections, which would skew the data. To help visualize the CNTM dataset, Figure 3 provides a magnified snapshot of t=60m. In this figure the nodes are on a path that is winding about a large hill, which blocks communication between many nodes on opposite sides. This is one example how the terrain may impact the connectivity more than the distance.

(a) CNTM t=0m   (b) CNTM t=30m   (c) CNTM t=60m   (d) CNTM t=90m   (e) CNTM t=120m

(f) CNCM t=0m   (g) CNCM t=30m   (h) CNCM t=60m   (i) CNCM t=90m   (j) CNCM t=120m
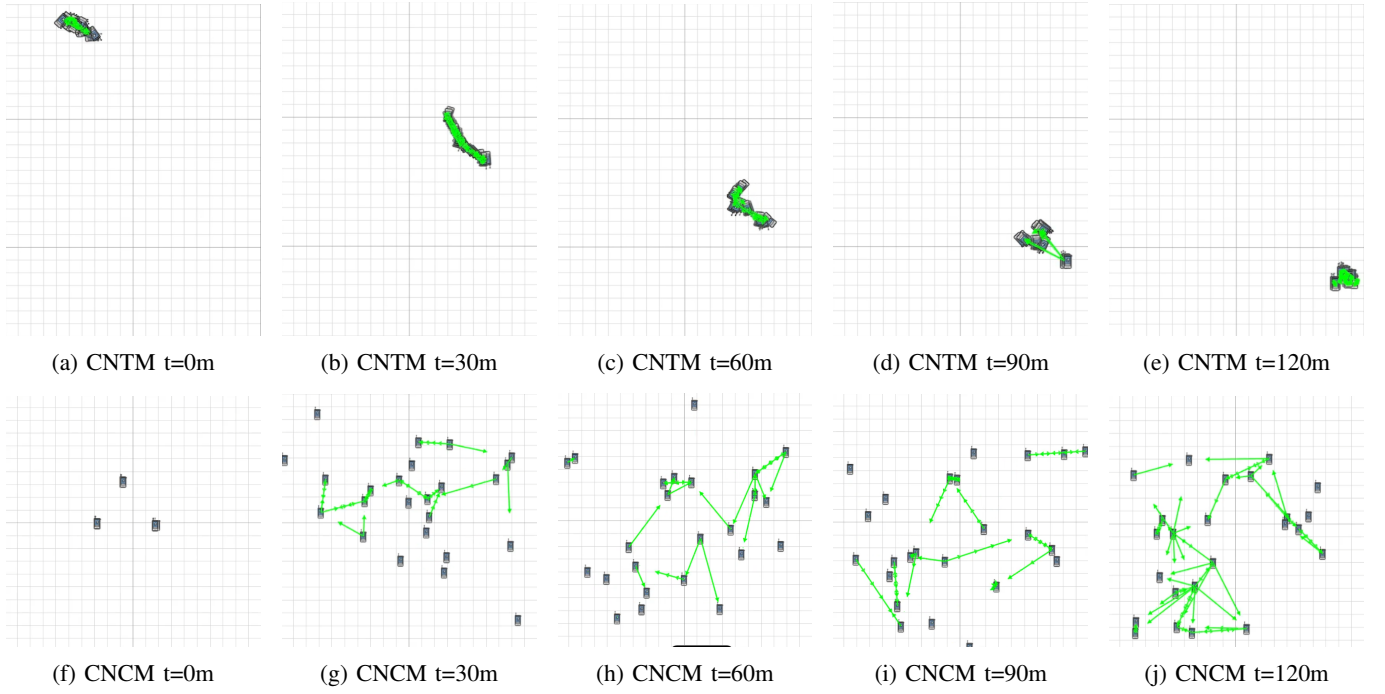
Fig. 2: Node positions over time (in minutes) for CNTM and CNCM datasets. The top row shows the objective-oriented mobility of the CNTM dataset, the tank company deploys from the upper left to bottom right. The bottom row shows the random mobility of the CNCM dataset. Nodes start off in 3 groups and randomly move toward and away from each other.
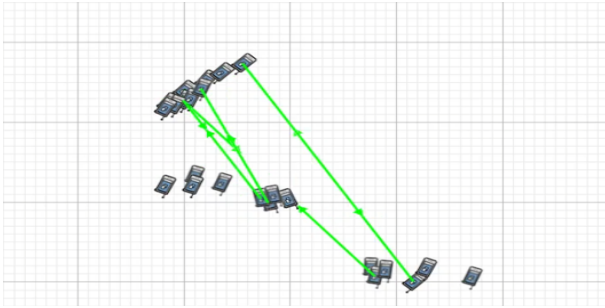

Fig. 3: CNTM t=60m magnified

The node connectivity and number of hops between nodes is displayed in Figure 4 for both the CNCM and CNTM datasets. We consider the number of hops as the minimum number of nodes required to send a message from one node to another, where nodes with direct connectivity are 1 hop away. In this figure, the x-axis refers to the destination node, and the y-axis refers to the source node. Note the diagonal line of white blocks in each subfigure where each node has no connectivity to itself. Additionally, there is some symmetry across this diagonal, since the path loss values are bidirectional. However, they are not perfectly symmetrical because the connectivity information updates when nodes send messages, and nodes do not send messages at the exact same time.

In Figure 4a, we see most nodes in the CNTM dataset start off as 1-hop neighbors, while a few nodes are 2 hops away. As time progresses, the connectivity becomes more sparse in Figure 4b, with some nodes being 4 hops apart, and others without any connectivity. Similarly with the CNCM

dataset, the overall connectivity is highest in Figure 4f, before becoming more sparse as nodes move away from each other in later time steps. The differences between the two datasets are represented in Figure 4, where the CNTM connections generally are more stable among groups of nodes. Nodes in CNTM are grouped in platoons and move in a coordinated manner, but these relationships are nonexistent for the CNCM dataset.

4) Data Processing: For both datasets, we split the data into varying time steps, namely, one-, two- and five-steps, as input. We label the connectivity by existence of the sender-receiver pairs, and with an observed threshold of 128 dBm in path loss, i.e., a pair of nodes are connected if their path loss is 128 dBm or below. For training input of non-graph models, we concatenate the sender-node features, receiver-node features and the edge features into one vector without pooling or compression.

Model	Architecture	Encoding
MLP	non-graph	general
LSTM	non-graph	temporal
GRU	non-graph	temporal
GCN	graph-based	spatial
GAT	graph-based	temporal
GATv2	graph-based	temporal
GTC	graph-based	temporal

TABLE II: Model Categorization

B. Baselines

We compare our proposed STGED framework with the following baselines:

(a) CNTM t=0m   (b) CNTM t=30m   (c) CNTM t=60m   (d) CNTM t=90m   (e) CNTM t=120m

(f) CNCM t=0m   (g) CNCM t=30m   (h) CNCM t=60m   (i) CNCM t=90m   (j) CNCM t=120m
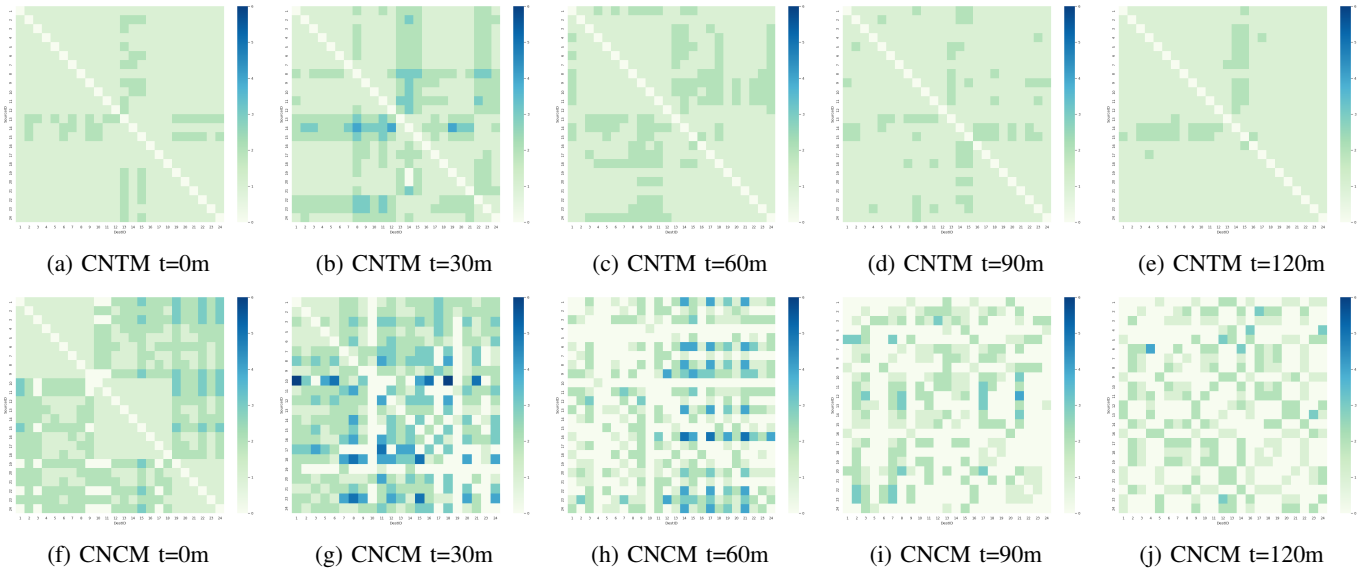
Fig. 4: Node hop count over time for the CNTM and CNCM datasets. This refers to the amount of nodes needed to transmit from the source node (y-axis) to the destination node (x-axis). If a node can send to another node directly, the hop count is 1. If a node has no route to another node, the hop count is 0 and the color is white. The top row of figures shows the number of hops needed to connect nodes of the CNTM dataset, and has a maximum of 4 hops at t=30m. The bottom row of figures illustrates number of hops needed to connect nodes of the CNCM dataset, and has a maximum of 6 hops at t=30m.

- **MLP** [40]: Multi-Layer Perceptron, which is a fully connected multi-layer neural network.
- **LSTM** [41]: Long Short-Term Memory, which is a recurrent neural network.
- **GRU** [42]: Gated Recurrent Unit, which is a recurrent neural network.
- **GCN** [43]: Graph Convolutional Network with Edge-Conditioned convolutional filters.
- **GAT** [44]: Graph Attention Network where its attention mechanism uses a single-layer feed-forward neural network.
- **GATv2** [45]: Graph Attention Network with improvement, where every node can attend to any other node.
- **GTC** [38]: Graph Transformer Convolutional network that uses scaled dot-product attention, similar to the original Transformer model.

Table II summarises the categories of the models in architectural and encoding types.

C. Experiment Setup

We split the datasets into 90:5:5 ratio for training, validation and test sets, respectively. As the baselines contains non-temporal models, we choose to split the data randomly instead of chronologically for fair evaluation.

To balance the positive and negative samples, we take the minimum of positive and negative sample sets, and randomly sample the same number of samples from the larger set in training time. In testing time, we retain all positive and negative samples for performance evaluation. We apply a 20% dropout to all components to further prevent over-fitting.

All models are trained on the CNTM dataset. The datasets are processed to have three different sizes of time steps, i.e., one-, two-, and five-step, as the input to the models to predict the next-state connectivity of the GTC. We measure the performance of the models with four metrics, namely accuracy (Acc), Precision (Pre), Recall (Rec) and F1-score (F1).

In terms of hyperparameters for STGED, the spatial encoder has two hidden layers with 1024 hidden units and 128 attention heads. The node embedding is of same size as the hidden units, i.e. 1024. The temporal encoder (i.e. LSTM) has two hidden layers and 2048 hidden units, which do not process compression to the two input node embeddings. All models runs over 20 epochs with a learning rate of 1e-6, and with additional epochs until they converge.

For fair comparison, all baselines are constructed with similar scale and trained with similar hyperparameters as compared to the proposed models.

All experiments are conducted on an Nvidia DGX server with 4 Tesla V100-DGXS-32GB GPUs. Each training or testing is conducted on a single GPU at a time.

D. Results

Table III shows the experimental results of the proposed STGED model and the six baseline methods. Below summarizes our observations and interpretations:

1) **Strong Performance of STGED**: This model consistently outperforms all baseline models across all steps and metrics. It exhibits both high precision and high recall, suggesting excellent performance in both identifying and not missing positive instances. Furthermore,

	1 Step				2 Steps				5 Steps			
	Acc	F1	Pre	Rec	Acc	F1	Pre	Rec	Acc	F1	Pre	Rec
MLP	0.868	0.921	0.929	0.938	0.910	0.947	0.970	0.995	0.920	0.935	0.943	0.951
GRU	0.783	0.855	0.835	0.815	0.893	0.915	0.955	0.998	0.918	0.925	0.959	0.996
LSTM	0.783	0.870	0.869	0.869	0.929	0.956	0.976	0.996	0.933	0.957	0.976	0.996
GCN	0.664	0.794	0.908	0.705	0.895	0.945	0.916	0.975	0.906	0.931	0.962	0.995
GAT	0.869	0.945	0.930	0.916	0.901	0.934	0.965	0.998	0.918	0.957	0.918	1.0
GATv2	0.906	0.951	0.956	0.953	0.916	0.956	0.978	1.0	0.936	0.985	0.992	1.0
GTC	0.906	0.951	0.956	0.953	0.916	0.956	0.978	1.0	0.936	0.985	0.992	1.0
STGED	**0.919**	**0.966**	**0.982**	**0.968**	**0.979**	**0.991**	**0.996**	**1.0**	**0.992**	**0.998**	**0.998**	**1.0**

TABLE III: Experimental Results for STGED and baselines for the next-state 1-hop connectivity prediction. Using input time steps of one, two and five, STGED consistently outperforms all baseline models across all steps and metrics.

Encoder	Accuracy	F1-score
GCN	0.503	0.780
GAT	0.508	0.423
GATv2	0.501	0.484
GTC	0.538	0.647
GCN-GRU	0.686	0.780
GAT-GRU	0.701	0.608
GATv2-GRU	0.713	0.685
GTC-GRU	0.720	0.682
GCN-LSTM	0.686	0.580
GAT-LSTM	0.708	0.623
GATv2-LSTM	0.723	0.668
GTC-LSTM	**0.739**	**0.720**

TABLE IV: Ablation Experimental Results

adding spatial encoding unit significantly improves the performance of the model across all steps.

2) **High Consistency in Attention-Based Graph Models**: GAT, GATv2 and TCN exhibit very consistent performance metrics across different steps. This implies that the attention mechanism is effective in learning temporal feature representation for this task.

3) **High Recall Across Models**: Almost all models, especially at two and five steps, have high or perfect recall. This is a significant trend that points to the models' ability to capture most of the positive instances. The high recall values could also indicate that the connectivity prediction task or the dataset makes it easier to capture positive instances. Given that different types of encoders (temporal, spatial, and general) also achieve high recall, this seems more a characteristic of the dataset or the task itself rather than the models.

E. Ablation Experiments

We conduct ablation experiment on proposed STGED framework to variate each component. For spatial encoder, we choose from: (1) GCN, (2) GAT, (3) GATv2 and (4) GTC. For temporal encoder, we choose between LSTM and GRU. Furthermore, we conduct further experimentations to assess effectiveness of temporal-encoder-only and spatial-encoder-only models. Finally, we use the CNCM dataset, i.e., random movements of the nodes without a goal, to benchmark the effectiveness in representation learning. All models use five time steps as input to predict the 1-hop connectivity of the following state.

Table IV shows the results of the ablation experiment. We observe that GTC-LSTM shows the highest accuracy and F1-

score. This aligns with our expectation because GTC uses dot-product attention among connected nodes, which has a global vision of latent spatial features; as for the temporal encoder, LSTM consists of more gates (i.e. input, output and forget gates) than the GRU (i.e. reset and update gates). While GRU is time-efficient in training, LSTM is more powerful and adaptable as a temporal encoder.

V. Conclusion

This paper presents a novel STGED framework for TCN representation learning. Two datasets demonstrate the ability of the STGED in learning latent features, such as the spatial-temporal tactical movements. We design experiment tasks to pass through varying network states (i.e. time steps) to the models and predict the next-state one-hop connectivity. Experimental results show that the STGED consistently outperforms all baseline models across all input steps and metrics, revealing advantages in STGED for representing TCNs.

References

[1] N. Suri, A. Hansson, J. Nilsson, P. Lubkowski, K. Marcus, M. Hauge, K. Lee, B. Buchin, L. Mısırhoğlu, and M. Peuhkuri, "A realistic military scenario and emulation environment for experimenting with tactical communications and heterogeneous networks," in *2016 International Conference on Military Communications and Information Systems (ICMCIS)*, 2016, pp. 1–8.

[2] B. L. Oshlag, D. Pecelli, and S. V. Pizzi, "An rsvp surrogate for guaranteed bandwidth for tactical communications," in *2006 IEEE Military Communications conference*, 2006, pp. 1–6.

[3] N. Suri, J. Nilsson, A. Hansson, U. Sterner, K. Marcus, L. Misirlioğlu, M. Hauge, M. Peuhkuri, B. Buchin, R. in't Velt, and M. Breedy, "The angloval tactical military scenario and experimentation environment," in *2018 International Conference on Military Communications and Information Systems (ICMCIS)*, 2018, pp. 1–8.

[4] R. Srinivasan and J. J. Garcia-Luna-Aceves, "TCP-RTA: Real-time Topology Adaptive Congestion Control Strategy in TCP," in *Ubiquitous Networking*, ser. Lecture Notes in Computer Science, E. Sabir, H. Elbiaze, F. Falcone, W. Ajib, and M. Sadik, Eds. Cham: Springer International Publishing, 2023, pp. 98–112.

[5] A. Singh, R. Kumar, and S. Pant, "Advanced routing on aodv using link prediction in mobile ad-hoc network," in *2017 3rd International Conference on Advances in Computing,Communication & Automation (ICACCA) (Fall)*, 2017, pp. 1–7.

[6] M. Peuhkuri, "start [Anglova]," Jan. 2022.

[7] N. Suri, K. M. Marcus, C. van den Broek, H. Bastiaansen, P. Lubkowski, and M. Hauge, "Extending the anglova scenario for urban operations," in *2019 International Conference on Military Communications and Information Systems (ICMCIS)*, 2019, pp. 1–7.

[8] Z. Liu, Q.-M. Zhang, L. Lü, and T. Zhou, "Link prediction in complex networks: A local naïve Bayes model," *Europhysics Letters*, vol. 96, no. 4, p. 48007, Nov. 2011.

[9] L. A. Adamic and E. Adar, "Friends and neighbors on the Web," *Social Networks*, vol. 25, no. 3, pp. 211–230, Jul. 2003.

[10] G. Berlusconi, F. Calderoni, N. Parolini, M. Verani, and C. Piccardi, "Link Prediction in Criminal Networks: A Tool for Criminal Intelligence Analysis," *PLOS ONE*, vol. 11, no. 4, p. e0154244, Apr. 2016, publisher: Public Library of Science.

[11] Z. Huang, X. Li, and H. Chen, "Link prediction approach to collaborative filtering," in *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, ser. JCDL '05. New York, NY, USA: Association for Computing Machinery, Jun. 2005, pp. 141–142.

[12] H. Shao, L. Wang, H. Liu, and R. Zhu, "A link prediction method for MANETs based on fast spatio-temporal feature extraction and LSGANs," *Scientific Reports*, vol. 12, no. 1, p. 16896, Oct. 2022, number: 1 Publisher: Nature Publishing Group.

[13] M. von Rechenberg, J. F. Loevenich, P. Zißner, P. H. L. Rettore, R. R. F. Lopes, and P. Sevenich, "Mobility prediction at the tactical edge: A handover for centralized/decentralized networks," in *2023 International Conference on Military Communications and Information Systems (ICM-CIS)*, 2023, pp. 1–8.

[14] J. Lopatka, T. Paso, R. Massin, and X. Leturc, "Multi band efficient networks for ad hoc communications," *Procedia Computer Science*, vol. 205, pp. 88–96, Jan. 2022.

[15] N. Suri, R. i. t. Velt, M. Breedy, E. Cramer, J. Nilsson, K. M. Marcus, A. Morelli, and R. Fronteddu, "To Forward or not to Forward: Considerations on and Experiments with Network Layer Forwarding in Combination with Middleware Services for Group Communications," *Procedia Computer Science*, vol. 205, pp. 78–87, Jan. 2022.

[16] B. Schütz, S. Thieme, C. Fuchs, D. Weber, and N. Aschenbruck, "Network Coding-based Multi-Path Forward Erasure Correction for Tactical Scenarios," in *2021 IEEE Military Communications Conference*, 11 2021, pp. 939–944, iSSN: 2155-7586.

[17] K. G. Preetha and A. Unnikrishnan, "Fuzzy Decision-Based Reliable Link Prediction Routing in MANET Using Belief Propagation," in *Proceedings of the Second International Conference on Computational Intelligence and Informatics*, ser. Advances in Intelligent Systems and Computing, V. Bhateja, J. M. R. Tavares, B. P. Rani, V. K. Prasad, and K. S. Raju, Eds. Singapore: Springer, 2018, pp. 427–434.

[18] S. Patel and H. Pathak, "A regression-based technique for link failure time prediction in manet," vol. 16, no. 2–3, 2020.

[19] Y. Maret, M. Raza, F. Legendre, J. Wang, N. Bessis, and J.-F. Wagen, "Investigation of a GNN approach to mitigate congestion in a realistic MANET scenario," *Procedia Computer Science*, vol. 205, no. C, pp. 127–136, Jan. 2022.

[20] A. Swaminathan, M. Chaba, D. K. Sharma, and U. Ghosh, "GraphNET: Graph Neural Networks for routing optimization in Software Defined Networks," *Computer Communications*, vol. 178, pp. 169–182, Oct. 2021.

[21] P. Sun, J. Lan, J. Li, Z. Guo, and Y. Hu, "Combining Deep Reinforcement Learning With Graph Neural Networks for Optimal VNF Placement," *IEEE Communications Letters*, vol. 25, no. 1, pp. 176–180, Jan. 2021, conference Name: IEEE Communications Letters.

[22] T. Chen, X. Zhang, M. You, G. Zheng, and S. Lambotharan, "A GNN-Based Supervised Learning Framework for Resource Allocation in Wireless IoT Networks," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 1712–1724, Feb. 2022, conference Name: IEEE Internet of Things Journal.

[23] Y. Guo, Y. Wang, F. Khan, A. A. Al-Atawi, A. A. Abdulwahid, Y. Lee, and B. Marapelli, "Traffic Management in IoT Backbone Networks Using GNN and MAB with SDN Orchestration," *Sensors*, vol. 23, no. 16, p. 7091, Jan. 2023, number: 16 Publisher: Multidisciplinary Digital Publishing Institute.

[24] J. Liu, K. L. Wood, and K. H. Lim, "Strategic and crowd-aware itinerary recommendation," in *Proceedings of the 2020 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD'20)*, 2020.

[25] J. Liu, T. Singhal, L. T. Blessing, K. L. Wood, and K. H. LIM, "Crisisbert: a robust transformer for crisis classification and contextual crisis embedding," in *Proceedings of the 32nd ACM Conference on Hypertext and Social Media*, 2021, pp. 133–141.

[26] J. Liu, Y. C. Ng, Z. Gui, T. Singhal, L. T. M. Blessing, K. L. Wood, and K. H. Lim, "Title2vec: a contextual job title embedding for occupational named entity recognition and other applications," *Journal of Big Data*, vol. 9, no. 1, pp. 1–16, 2022.

[27] J. Liu, Y. C. Ng, K. L. Wood, and K. H. Lim, "Ipod: A large-scale industrial and professional occupation dataset," in *Proceedings of the 2020 ACM Conference on Computer Supported Cooperative Work and Social Computing Companion (CSCW'20)*, 2020, pp. 323–328.

[28] Y. George, S. Karunasekera, A. Harwood, and K. H. Li, "Spatio-temporal event detection using poisson model and quad-tree on geotagged social media," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 2247–2256.

[29] Y. George, S. Karunasekera, A. Harwood, and K. H. Lim, "Real-time spatio-temporal event detection on geotagged social media," *Journal of Big Data*, vol. 8, no. 1, p. 91, 2021.

[30] T. Singhal, J. Liu, K. H. Lim, and L. T. Blessing, "Analyzing scientific publications using domain-specific word embedding and topic modelling," in *International Journal of Data Mining Science*, 2021.

[31] J. Heng, J. Liu, and K. H. Lim, "Urban crowdsensing using social media: An empirical study on transformer and recurrent neural networks," in *Proceedings of the 2020 IEEE International Conference on Big Data*, 2020.

[32] M. Li, K. H. Lim, T. Guo, and J. Liu, "A transformer-based framework for poi-level social post geolocation," in *European Conference on Information Retrieval*. Springer Nature Switzerland Cham, 2023, pp. 588–604.

[33] G. Chen, Y. Guo, Q. Zeng, and Y. Zhang, "A Novel Cellular Network Traffic Prediction Algorithm Based on Graph Convolution Neural Networks and Long Short-Term Memory through Extraction of Spatial-Temporal Characteristics," *Processes*, vol. 11, no. 8, p. 2257, Aug. 2023, number: 8 Publisher: Multidisciplinary Digital Publishing Institute.

[34] Z. Wang, J. Hu, G. Min, Z. Zhao, Z. Chang, and Z. Wang, "Spatial-Temporal Cellular Traffic Prediction for 5G and Beyond: A Graph Neural Networks-Based Approach," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 4, pp. 5722–5731, Apr. 2023, conference Name: IEEE Transactions on Industrial Informatics.

[35] S. Lan, Y. Ma, W. Huang, W. Wang, H. Yang, and P. Li, "DSTAGNN: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 11 906–11 917.

[36] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, pp. 4189–4196, May 2021.

[37] S. Li, W. Meng, S. He, J. Bi, and G. Liu, "Staged: A spatial-temporal aware graph encoder-decoder for fault diagnosis in industrial processes," *IEEE Transactions on Industrial Informatics*, 2023.

[38] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked label prediction: Unified message passing model for semi-supervised classification," 2021.

[39] Keysight, "EXata Network Modeling," section: Article Section.

[40] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

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

[42] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014.

[43] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International conference on machine learning*. PMLR, 2017, pp. 1263–1272.

[44] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio *et al.*, "Graph attention networks," *stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.

[45] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" *arXiv preprint arXiv:2105.14491*, 2021.