ADVERSPARSE: AN ADVERSARIAL ATTACK FRAMEWORK FOR DEEP SPATIAL-TEMPORAL GRAPH NEURAL NETWORKS

Jiayu Li[⋆] *Tianyun Zhang*[†] *Shengmin Jin*[⋆] *Makan Fardad*[⋆] *Reza Zafarani*[⋆]

[⋆] Syracuse University [†]Cleveland State University

ABSTRACT

Spatial-temporal graph have been widely observed in various domains such as neuroscience, climate research, and transportation engineering. The state-of-the-art models of spatial-temporal graphs rely on Graph Neural Networks (GNNs) to obtain explicit representations for such networks and to discover hidden spatial dependencies in them. These models have demonstrated superior performance in various tasks. In this paper, we propose a sparse adversarial attack framework ADVERSPARSE to illustrate that when only a few key connections are removed in such graphs, hidden spatial dependencies learned by such spatial-temporal models are significantly impacted, leading to various issues such as increasing prediction errors. We formulate the adversarial attack as an optimization problem and solve it by the Alternating Direction Method of Multipliers (ADMM). Experiments show that ADVERSPARSE can find and remove key connections in these graphs, leading to malfunctioning models, even in models capable of learning hidden spatial dependencies.

Index Terms— Graph sparsification, adversarial attack

1. INTRODUCTION

Spatial-temporal graphs are universal in many domains, especially in transportation. For example, traffic forecasting using spatial-temporal graphs is an extremely challenging task due to complex spatial dependencies and varying temporal trends. Recent studies have captured the spatial dependencies by using Graph Neural Networks (GNNs), while Recurrent Neural Networks (RNNs) are used to model the temporal trends [1, 2, 3, 4]. To improve the performance of tasks such as traffic forecasting, developing GNN models for spatial-temporal graphs has been in the spotlight. For example, Graph WaveNet [5] illustrates that traditional GNNs only capture the explicit graph structure and implicit relations may be missed due to incomplete information (i.e., missing connections). To obtain explicit spatial dependencies, Graph WaveNet uses graph diffusion convolution $\mathbf{Z} = \sum_{k=0}^{K} \mathbf{P}^k \mathbf{X}$, where $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$, $\mathbf{D}$ is the diagonal degree matrix of an adjacency matrix $\mathbf{A}$ of a graph with N nodes and $|E|$ edges, and $\mathbf{X}$ is an input graph signal. To discover implicit or hidden connections, Graph WaveNet proposes to learn a self-adaptive adjacency matrix $\widetilde{\mathbf{A}}_{apt} = \text{SoftMax}(\text{ReLU}(\mathbf{E}_1\mathbf{E}_2^T))$, which can be considered as the transition matrix of a hidden diffusion process. The learnable parameters $\mathbf{E}_1, \mathbf{E}_2 \in \mathbf{R}^{N\times H}$ can be randomly initialized. Finally, Graph WaveNet defines $\mathbf{Z} = \sum_{k=0}^{K} \mathbf{P}^k\mathbf{X} + \widetilde{\mathbf{A}}_{apt}^k\mathbf{X}$ to capture both explicit and hidden spatial dependencies.

Recent studies have shown that GNNs are vulnerable to adversarial attacks [6, 7, 8, 9, 10]. Attackers can generate adversarial perturbations by manipulating the graph structure. Here, we develop attacks for spatial-temporal GNN models, especially those that have the capability to resist such attacks, e.g., by estimating hidden connections. In particular, we attack state-of-the-art Graph WaveNet, where we only remove a few connections (i.e. perform *sparsification* [11]) on the filter related to $\mathbf{A}$ in the graph diffusion convolution. This is similar to a traffic jam happening in a road network because a few important roads are blocked. Our attack ensures that the learnable $\widetilde{\mathbf{A}}_{apt}$ cannot recover the removed connections so that the model performance cannot be maintained.

We first analyze the relationship between sparsification and graph diffusion convolution. We can state the transition matrix $\mathbf{P}$ as $\mathbf{P} = \mathbf{U}\Lambda\mathbf{U}^{-1}$ and the graph signal as $\mathbf{X} = \mathbf{U}\mathbf{C}$, where $\mathbf{U} = (u_1, \ldots, u_N)$ is the basis signal and $\mathbf{C} = (c_1, \ldots, c_N)$ is the coefficients for $\mathbf{U}$. Hence, we can have

$$\begin{aligned}\mathbf{Z} &= \sum_{k=0}^{K} \mathbf{P}^k\mathbf{X} \\ &= \mathbf{U}\Lambda^0\mathbf{U}^{-1}\mathbf{U}\mathbf{C} + \mathbf{U}\Lambda^1\mathbf{U}^{-1}\mathbf{U}\mathbf{C} + \cdots + \mathbf{U}\Lambda^K\mathbf{U}^{-1}\mathbf{U}\mathbf{C} \\ &= \sum_{1\le i\le N} \lambda_i^0 c_i u_i + \sum_i \lambda_i^1 c_i u_i + \cdots + \sum_i \lambda_i^K c_i u_i \\ &= \sum_{1\le i\le N} (\lambda_i^0 + \lambda_i^1 + \cdots + \lambda_i^K) c_i u_i.\end{aligned} \tag{1}$$

In Eq. 1, a filter function can be defined as $h(\lambda_i) = \sum_{k=0}^{K} \lambda_i^k$. As the eigenvalues λ_i of $\mathbf{P}$ are all bounded above by 1, we can use geometric series to derive a closed-form expression for $h(\lambda_i) = \frac{1}{1-\lambda_i}$. It is easy to find the relationship between the eigenvalues of $\mathbf{P}$ and eigenvalues (μ_i's) of a normalized Laplacian matrix $\mathbf{L}$: $\mu_i = 1 - \lambda_i$. Therefore, we can restate the filter function corresponding to eigenvalues μ_i, i.e. $h(\mu_i) = \frac{1}{\mu_i}$. The number of 0 in the eigenvalues of matrix $\mathbf{L}$ denotes the number of connected components and sparsifica-

tion can increase the number of connected components.

In this paper, we propose ADVERSPARSE, a sparsification framework to attack spatial-temporal graph models. ADVERSPARSE achieves its objective by generating an adversarial perturbation Δ on the adjacency matrix $\mathbf{A}$. We show that only a few important connections need to be removed for the attack to succeed. We specifically target Graph WaveNet, as the state-of-the-art, to showcase the efficiency of the proposed framework. The performance drop of Graph WaveNets due to the attack indicates that the negative effect of removed edges cannot be mitigated by Graph WaveNet's learnable self-adaptive adjacency matrix $\widetilde{\mathbf{A}}_{apt}$. Our experiments demonstrate that compared to other attacking strategies including random noise, degree-based noise, and PageRank-based noise, ADVERSPARSE not only removes fewer edges, but the performance of Gragh WaveNet also drops the most.

2. ADVERSPARSE: ADVERSARIAL ATTACK ON DEEP SPATIAL-TEMPORAL GRAPH MODELS

2.1. Problem Formulation

In Spatial-temporal graph modeling, the output of the graph neural network [5] is

$$\hat{\mathbf{X}}^{(t+1):(t+T)} = f(\mathbf{X}^{(t-S):t}, \mathbf{A}; \mathbf{\Theta}), \tag{2}$$

where $\mathbf{X}^{(t-S):t} \in \mathbf{R}^{N\times D\times S}$ denotes the historical S-step graph signals, $\mathbf{A} \in \mathbf{R}^{N\times N}$ denotes the adjacency matrix of a given graph, f is the mapping relation learned by the graph neural network to forecast the next T-step graph signals when the historical S-step graph signals is given, and $\mathbf{\Theta}$ is the collection of parameters in the graph neural network.

Graph neural network training aims to minimize the Mean Absolute Error (MAE) of the predicted and real graph signals in the next T steps. The training loss is defined by

$$L(\mathbf{X}^{(t-S):(t+T)}, \mathbf{A}; \mathbf{\Theta}) = \frac{1}{TND}\sum_{i=1}^{T}\sum_{j=1}^{N}\sum_{k=1}^{D}|\hat{\mathbf{X}}_{jk}^{(t+i)} - \mathbf{X}_{jk}^{(t+i)}| \tag{3}$$

Hereafter, for simplicity of notation, we present the training loss as $L(\mathbf{X}, \mathbf{A}; \mathbf{\Theta})$.

In the adversarial attack on deep spatial-temporal graph modeling, we propose to increase the MAE of the prediction produced by the graph neural network by removing limited links in the graph. We formulate this problem as

$$\begin{aligned} \underset{\mathbf{\Delta}}{\text{maximize}} \quad & L(\mathbf{X}, \mathbf{A} - (\mathbf{A} - \mathbf{I}) \circ \mathbf{\Delta}; \mathbf{\Theta}) \\ \text{subject to} \quad & \mathbf{\Delta} \in \{0,1\}^N, \|\mathbf{\Delta}\|_0 \leq l, \end{aligned} \tag{4}$$

where $\circ$ denotes element-wise multiplication, $\mathbf{\Delta}$ is the optimization variable in this problem, l denotes the limited number of links that can be removed, and $\mathbf{I}$ is the identity matrix. Both $\mathbf{\Delta}$ and $\mathbf{I}$ have the same size as $\mathbf{A}$. The matrix $(\mathbf{A} - (\mathbf{A} - \mathbf{I}) \circ \mathbf{\Delta})$ denotes the adjacency matrix of the graph after some links are removed.

2.2. Proposed Solution

Problem (4) can be equivalently rewritten as

$$\begin{aligned} \underset{\mathbf{\Delta}}{\text{minimize}} \quad & -L(\mathbf{X}, \mathbf{A} - (\mathbf{A} - \mathbf{I}) \circ \mathbf{\Delta}; \mathbf{\Theta}) \\ \text{subject to} \quad & \mathbf{\Delta} \in \xi, \end{aligned} \tag{5}$$

where the set $\xi = \{\mathbf{\Delta} \mid \mathbf{\Delta} \in \{0,1\}^N, \|\mathbf{\Delta}\|_0 \leq l\}$. In this problem, the l_0 and binary constraints are non-convex, thus the set ξ is non-convex. In general, it is difficult to solve the problem with non-convex constraints, motivating us to propose an algorithm to solve problem (5).

Problem (5) can be equivalently rewritten in a constraint-free format, which is

$$\underset{\mathbf{\Delta}}{\text{minimize}} \quad -L(\mathbf{X}, \mathbf{A} - (\mathbf{A} - \mathbf{I}) \circ \mathbf{\Delta}; \mathbf{\Theta}) + g(\mathbf{\Delta}), \tag{6}$$

where $g(\cdot)$ is the indicator function for set ξ, defined as

$$g(\mathbf{\Delta}) = \begin{cases} 0 & \text{if } \mathbf{\Delta} \in \xi, \\ +\infty & \text{otherwise.} \end{cases}$$

In problem (6), the first term is the negative training loss of the graph neural network, which is differentiable. While the second term is the non-differentiable indicator function, which results in the objective function for problem (6) to be non-differentiable. It is difficult to solve problem (6) directly, while recent research has shown that such problems can be solved efficiently by the ADMM [12]. We first equivalently rewrite problem (6) in ADMM form as

$$\begin{aligned} \underset{\mathbf{\Delta}}{\text{minimize}} \quad & -L(\mathbf{X}, \mathbf{A} - (\mathbf{A} - \mathbf{I}) \circ \mathbf{\Delta}; \mathbf{\Theta}) + g(\mathbf{\Omega}) \\ \text{subject to} \quad & \mathbf{\Delta} = \mathbf{\Omega}, \end{aligned} \tag{7}$$

The augmented Lagrangian [12] of problem (7) is given by

$$\begin{aligned} \ell_\rho(\mathbf{X}, \mathbf{A}, \mathbf{\Delta}, \mathbf{\Omega}, \mathbf{\Lambda}; \mathbf{\Theta}) = -L(\mathbf{X}, \mathbf{A} - (\mathbf{A} - \mathbf{I}) \circ \mathbf{\Delta}; \mathbf{\Theta}) + \\ g(\mathbf{\Omega}) + \text{tr}[\mathbf{\Lambda}^T(\mathbf{\Delta} - \mathbf{\Omega})] + \frac{\rho}{2}\|\mathbf{\Delta} - \mathbf{\Omega}\|_F^2, \end{aligned}$$

where $\mathbf{\Lambda}$ is the dual variable or Lagrange multiplier, ρ is the penalty parameter, $\text{tr}(\cdot)$ denotes the trace of a matrix, and $\|\cdot\|_F^2$ denotes the Frobenius norm.

The augmented Lagrangian can be equivalently rewritten in the scaled form, which is

$$\begin{aligned} \ell_\rho(\mathbf{X}, \mathbf{A}, \mathbf{\Delta}, \mathbf{\Omega}, \mathbf{U}; \mathbf{\Theta}) = -L(\mathbf{X}, \mathbf{A} - (\mathbf{A} - \mathbf{I}) \circ \mathbf{\Delta}; \mathbf{\Theta}) + \\ g(\mathbf{\Omega}) + \frac{\rho}{2}\|\mathbf{\Delta} - \mathbf{\Omega} + \mathbf{U}\|_F^2 - \frac{\rho}{2}\|\mathbf{U}\|_F^2, \end{aligned}$$

where $\mathbf{U} = (1/\rho)\mathbf{\Lambda}$ is the scaled dual variable. We can perform ADMM [12] steps as

$$\mathbf{\Delta}^{k+1} := \underset{\mathbf{\Delta}}{\arg\min} \quad \ell_\rho(\mathbf{X}, \mathbf{A}, \mathbf{\Delta}, \mathbf{\Omega}^k, \mathbf{U}^k; \mathbf{\Theta}) \tag{8}$$

$$\mathbf{\Omega}^{k+1} := \underset{\mathbf{\Omega}}{\arg\min} \quad \ell_\rho(\mathbf{X}, \mathbf{A}, \mathbf{\Delta}^{k+1}, \mathbf{\Omega}, \mathbf{U}^k; \mathbf{\Theta}) \tag{9}$$

$$\mathbf{U}^{k+1} := \mathbf{U}^k + \mathbf{\Delta}^{k+1} - \mathbf{\Omega}^{k+1} \tag{10}$$

until

$$\|\mathbf{\Delta}^{k+1} - \mathbf{\Omega}^{k+1}\|_F^2 \leq \epsilon, \quad \|\mathbf{\Omega}^{k+1} - \mathbf{\Omega}^{k}\|_F^2 \leq \epsilon. \tag{11}$$

In problem (8), we solve

$$\underset{\mathbf{\Delta}}{\text{minimize}} \quad -L(\mathbf{X}, \mathbf{A} - (\mathbf{A} - \mathbf{I}) \circ \mathbf{\Delta}; \mathbf{\Theta}) + \frac{\rho}{2}\|\mathbf{\Delta} - \mathbf{\Omega}^k + \mathbf{U}^k\|_F^2. \tag{12}$$

In this problem, both the training loss and the Frobenius norm are differentiable; thus, we can use gradient descent to solve it. In problem (9), we solve

$$\underset{\mathbf{\Omega}}{\text{minimize}} \quad g(\mathbf{\Omega}) + \frac{\rho}{2}\|\mathbf{\Delta}^{k+1} - \mathbf{\Omega} + \mathbf{U}^k\|_F^2. \tag{13}$$

In this problem, the first term is the indicator function of the set ξ, and the second term is the Frobenius norm. In this case, the problem can be solved analytically. The solution is

$$\mathbf{\Omega}^{k+1} = \mathbf{\Pi}_\xi(\mathbf{\Delta}^{k+1} + \mathbf{U}^k). \tag{14}$$

The set ξ is the combination of two non-convex constraint sets, and in general it is difficult to derive the Euclidean projection onto the combination of non-convex sets. In Proposition 1, we show that the Euclidean projection onto the set ξ has a closed-form solution.

Proposition 1. *Given* $\xi = \{\mathbf{\Delta} \mid \mathbf{\Delta} \in \{0,1\}^N, \|\mathbf{\Delta}\|_0 \leq l\}$, *the Euclidean projection of a matrix* $\mathbf{Y}$ *onto the set* ξ *is*

$$(\mathbf{\Pi}_\xi(\mathbf{Y}))_{ij} = \begin{cases} 0 & \text{if } (\mathbf{Y})_{ij} < \max\{0.5, r^{(l)}\}; \\ 1 & \text{otherwise,} \end{cases}$$

where $(\cdot)_{ij}$ *is the element in the* i*-th row and* j*-th column of a matrix, and* $r^{(l)}$ *denotes the* l*-th largest element in* $\mathbf{\Delta}$.

Proof. When calculating $\mathbf{\Pi}_\xi(\mathbf{Y})$, we project $\mathbf{Y}$ onto a point in the set ξ with the closest Euclidean distance. The set ξ requires us to project every element to 0 or 1, and at most l elements are 1 after the projection. We define the l-th largest element in $\mathbf{\Delta}$ as $r^{(l)}$ and project the elements which are smaller than $r^{(l)}$ to 0. The reason is that smaller elements have closer Euclidean distance to 0 rather than 1. Hence, the l_0 constraint is satisfied, and for the elements which are greater than or equal to $r^{(l)}$, we consider the following two cases:

1) When $r^{(l)} < 0.5$, we project the elements in the range of $r^{(l)}$ and 0.5 to 0 and project the other elements to 1. This results in the closest Euclidean projection.

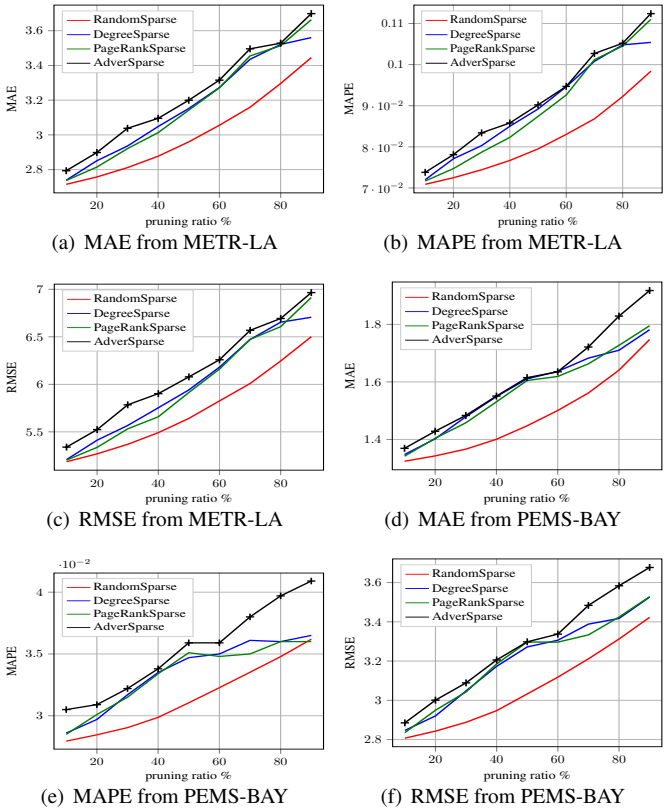
2) When $r^{(l)} > 0.5$, for the elements which are greater than or equal to $r^{(l)}$, we project these elements to 1 as they have a closer distance to 1 rather than 0.

In sum, when $r^{(l)} < 0.5$, we project the elements that are smaller than 0.5 to 0, and project the other elements to 1. And when $r^{(l)} > 0.5$, we project the elements that are smaller than $r^{(l)}$ to 0, and project the other elements to 1. Hence, we project an element to 0 if it is smaller than $\max\{0.5, r^{(l)}\}$ and otherwise, we project it to 1. □

Table 1. Dataset statistics

Dataset	# Nodes	# Edges	# Time Steps
METR-LA	207	1,515	34,272
PEMS-BAY	325	2,369	52,116

Finally, we update the scaled dual variable $\mathbf{U}$ according to (10). This concludes one iteration of ADMM. We iteratively update the variables until convergences (Eq. 11).



(a) MAE from METR-LA (b) MAPE from METR-LA

(c) RMSE from METR-LA (d) MAE from PEMS-BAY

(e) MAPE from PEMS-BAY (f) RMSE from PEMS-BAY

Fig. 1. We vary the pruning ratio from 10% to 90% with a 10% step-size to compare the performance of **AdverSparse**, DegreeSparse, PageRankSparse and RandomSparse on Graph WaveNet for 30-minute-ahead prediction.

3. EXPERIMENTS

In our experiments, we verify the effectiveness of ADVERSPARSE on Graph WaveNet by using two public traffic network datasets: METR-LA and PEMS-BAY [1]. Data statistics are in Table 1. METR-LA uses 207 sensors to record four months of data on traffic speed on the highways of Los Angeles County, while PEMS-BAY uses 325 sensors in the Bay area to obtain six months of traffic speed information. The readings of the sensors are aggregated into 5-minutes

Table 2. Performance comparison of different sparsification attacks on Graph WaveNet with smaller adversarial perturbations (i.e. pruning ratio $p = 10\%$). Attacks by ADVERSPARSE lead to the worst traffic forecasting results on both datasets.

			15min			30min			60min		
Dataset	Models	Attack method	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
METR-LA	Graph WaveNet	(no attack)	2.436	0.060	4.479	2.682	0.070	5.117	3.049	0.081	6.070
		RandomSparse	2.484	0.061	4.561	2.715	0.071	5.186	3.088	0.082	6.144
		DegreeSparse	2.515	0.062	4.593	2.739	0.072	5.208	3.108	0.083	6.174
		PageRankSparse	2.504	0.062	4.575	2.738	0.072	5.203	3.102	0.082	6.150
		ADVERSPARSE	**2.519**	**0.063**	**4.631**	**2.794**	**0.074**	**5.340**	**3.143**	**0.084**	**6.224**
PEMS-BAY	Graph WaveNet	(no attack)	1.091	0.022	2.162	1.310	0.028	2.782	1.570	0.035	3.506
		RandomSparse	1.102	0.022	2.181	1.324	0.028	2.807	1.582	0.036	3.524
		DegreeSparse	1.111	0.022	2.196	1.348	0.029	2.847	1.604	0.036	3.570
		PageRankSparse	1.112	0.022	2.198	1.342	0.029	2.835	1.592	0.036	3.537
		ADVERSPARSE	**1.125**	**0.023**	**2.212**	**1.369**	**0.031**	**2.885**	**1.611**	**0.037**	**3.581**

windows. We use the following attacks on Graph WaveNet as baselines to compare with ADVERSPARSE. [1]

- *RandomSparse (random noise)*, where we randomly prune ratio p of edges in the adjancency matrix, i.e., removing $\lfloor p \times |E| \rfloor$ edges. The probability that each edge is removed is $\frac{1}{|E|}$.
- *DegreeSparse (degree-based noise)*, where we remove edges between high-degree nodes. This can significantly impact natural connectivity of the graph, reducing its robustness [13]. We sort the nodes based on their degrees and remove $\lfloor p \times |E| \rfloor$ connections between high-degree nodes, where p is the pruning ratio.
- *PageRankSparse (PageRank-based noise)*. The PageRank scores for the nodes can reflect their importance [14]. We compute the PageRank score for each node and sort the nodes based on their scores. We remove $\lfloor p \times |E| \rfloor$ connections after sorting the scores, where p is the pruning ratio.

Experiments are conducted on a computer with one Intel(R) Xeon(R) 6248R CPU @ 3.00GHz and one Quadro RTX 6000 GPU card. The settings of parameters in all methods are the same as the default settings of Graph WaveNet. The pruning ratio p is varied from 10% to 90% in 10% intervals for all methods. When we calculate the Euclidean projection in Proposition 1, $l = \lfloor p \times |E| \rfloor$. We set the ρ as 0.5 in ADVERSPARSE. The evaluation metrics include mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE).

3.1. Experimental Results

In Table 2, 10% of the connections are removed (i.e., a small amount of noise). The results show that ADVERSPARSE leads to higher prediction errors in Graph WaveNet, when compared to other methods for 15-minute-, 30-minute-, and 60-minutes-ahead predictions on both datasets.

In Figure 1, we compare the MAE, MAPE and RMSE for 30-minutes-ahead prediction for different attack methods with 10% to 90% pruning ratios. The attacks from DegreeSparse and PageRankSparse are based on graph structure, and they have similar performances on Graph WaveNet. The attack by RandomSparse is the least effective on Graph WaveNet. ADVERSPARSE consistently leads to the highest errors for Graph WaveNet on both datasets for all pruning ratios. For example, Fig 1 illustrates that when compared to other baselines, ADVERSPARSE requires less noise (corresponding to a smaller pruning ratio) to achieve the same adversarial effect on the performance of Graph WaveNet.

4. CONCLUSION

Graph WaveNet is the state-of-the-art for spatial-temporal graph modeling. In Graph WaveNet, an adjacency matrix can capture explicit spatial dependencies while a learnable self-adaptive adjacency matrix can discover and model the spatial hidden relations. In this paper, we propose ADVERSPARSE to illustrate that when key connections are removed in networks, the hidden spatial dependencies learned by Graph WaveNet cannot mitigate the loss of such connections; hence, increasing prediction errors. Experimental results on real-world datasets demonstrate that ADVERSPARSE leads to higher prediction errors on Graph WaveNet compared to random noise, degree-based noise, and PageRank-based noise.

5. ACKNOWLEDGEMENTS

This research was supported in part by the National Science Foundation under awards CAREER IIS-1942929, CAREER CMMI-1750531, and ECCS-1609916.

[1] We cannot consider the spectral sparsification method here as the number of removal edge cannot be controlled. All codes are released at `https://github.com/Code4Graph/ADVERSPARSE`.

6. REFERENCES

[1] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018.

[2] Mengzhang Li and Zhanxing 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.

[3] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 922–929, Jul. 2019.

[4] Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu, "Traffic flow prediction via spatial temporal graph neural network," in *Proceedings of The Web Conference 2020*, New York, NY, USA, 2020, WWW '20, p. 1082–1092, Association for Computing Machinery.

[5] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2019, IJCAI'19, p. 1907–1913, AAAI Press.

[6] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song, "Adversarial attack on graph structured data," in *Proceedings of the 35th International Conference on Machine Learning*, Jennifer Dy and Andreas Krause, Eds. 10–15 Jul 2018, vol. 80 of *Proceedings of Machine Learning Research*, pp. 1115–1124, PMLR.

[7] Daniel Zügner and Stephan Günnemann, "Adversarial attacks on graph neural networks via meta learning," in *International Conference on Learning Representations*, 2019.

[8] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann, "Adversarial attacks on neural networks for graph data," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, New York, NY, USA, 2018, KDD '18, p. 2847–2856, Association for Computing Machinery.

[9] Han Xu, Yaxin Li, Wei Jin, and Jiliang Tang, "Adversarial attacks and defenses: Frontiers, advances and practice," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, New York, NY, USA, 2020, KDD '20, p. 3541–3542, Association for Computing Machinery.

[10] Wei Jin, Yaxing Li, Han Xu, Yiqi Wang, Shuiwang Ji, Charu Aggarwal, and Jiliang Tang, "Adversarial attacks and defenses on graphs," *SIGKDD Explor. Newsl.*, vol. 22, no. 2, pp. 19–34, Jan. 2021.

[11] Jiayu Li, Tianyun Zhang, Hao Tian, Shengmin Jin, Makan Fardad, and Reza Zafarani, "Graph sparsification with graph convolutional networks," *International Journal of Data Science and Analytics*, pp. 1–14, 2021.

[12] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[13] Jun Wu, Mauricio Barahona, Yue-Jin Tan, and Hong-Zhong Deng, "Spectral measure of structural robustness in complex networks," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 41, no. 6, pp. 1244–1252, 2011.

[14] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, "The pagerank citation ranking: Bringing order to the web.," Technical Report 1999-66, Stanford InfoLab, November 1999.