TIME DELAY ESTIMATION OF TRAFFIC CONGESTION BASED ON STATISTICAL CAUSALITY

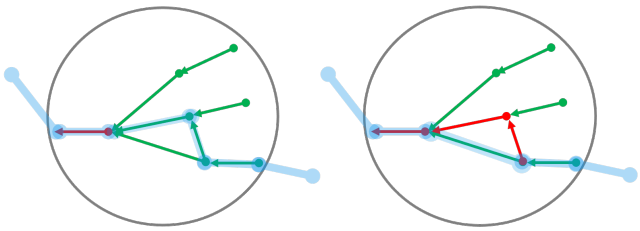
Anonymous authors
Paper under double-blind review

ABSTRACT

Considering how congestion will propagate in the near future, understanding traffic congestion propagation has become crucial in GPS navigation systems for providing users with a more accurate estimated time of arrival (ETA). However, providing the exact ETA during congestion is a challenge owing to the complex propagation process between roads and high uncertainty regarding the future behavior of the process. To aid in accurate ETA calculation during congestion, we propose a novel time delay estimation method for the propagation of traffic congestion due to traffic accidents using lag-specific transfer entropy (TE). Nonlinear normalization with a sliding window is used to effectively reveal the causal relationship between the source and target time series in calculating the TE. Moreover, Markov bootstrap techniques were adopted to quantify the uncertainty in the time delay estimator. To the best of our knowledge, the proposed method is the first to estimate the time delay based on the causal relationship between adjacent roads. The proposed method was validated using simulated data as well as real user trajectory data obtained from a major GPS navigation system applied in South Korea.

1 INTRODUCTION

Traffic congestion has been a universal problem for urban cities owing to the dramatic growth of population and the corresponding increase in vehicles, the economy, infrastructure, and proliferation of delivery services, among other factors. Traffic congestion affects nearby roads and causes additional congestion, particularly in all traffic leading to congested roads (Nguyen et al., 2016), causing greater damage to the traffic network.



(a) Road conditions when the driver requests ETA

(b) Road conditions when the driver arrives on the road

Figure 1: Motivating example: GPS navigation system recommending a travel route (blue lines) based on the related road conditions (red segment, congested; green segment, free flow).

Understanding traffic congestion propagation has become crucial in GPS navigation systems to provide users with a more accurate estimated time of arrival (ETA), considering how congestion propagation patterns in the near future. However, it is challenging to provide an exact ETA under cases of congestion owing to the complex propagation process between roads and high uncertainty about the future behavior of the process. Figure 1 illustrates the motivation for this study. Without understanding traffic congestion propagation, the GPS navigation system suggests the route (the blue line) based on the road conditions when the driver requests an ETA, as shown in Figure 1(a). However, the suggested route in Figure 1(a) faces severe traffic congestion when the driver arrives in a congested area because congestion propagates while the driver is moving. If the GPS navigation system

can identify the time delays between roads owing to congestion propagation, then it would suggest a different route, as shown in the blue line in Figure 1(b). Hence, to provide an accurate ETA in urban cities, a time delay estimation of traffic congestion propagation is inevitable.

In this paper, we propose a new time delay estimation method for traffic congestion propagation between roads using lag-specific Transfer Entropy (TE). Our main contributions are as follows:

- We provide a model-free approach to estimate congestion propagation delays using a lag-specific TE estimator in complex urban road systems.
- We show that a decomposition and a nonlinear normalization with a sliding window are effective time series preprocessing methods in revealing the causal relationship between traffic speed data.
- We quantify uncertainty in time delay estimation using bootstrap techniques. This uncertainty quantification allows us to evaluate the reliability of time delay estimates and serves as a basis for optimal hyperparameter tuning.
- We validate the proposed method through numerical simulations as well as real user trajectory data obtained from one of the major GPS navigation systems in South Korea.

This paper is organized as follows: Section 2 reviews related studies. Section 3 describes the background necessary to understand the proposed method. Section 4 proposes our new time delay estimation method. Sections 5 and 6 validate the proposed method using simulated data and real congestion propagation data in a road network, respectively. Finally, concluding remarks are made in Section 7.

2 Related Work

Congestion Propagation Pattern Analysis Recent studies on congestion propagation patterns have been widely conducted, focusing on an analysis of spatio-temporal patterns. The STOTree and frequent subtree algorithms were proposed based on outlier detection approaches for each road segment (Liu et al., 2011). Propagation Graphs were used to predict patterns of congestion propagation (Xiong et al., 2018). However, existing works on congestion propagation patterns mostly focus on detecting frequent congestion propagation patterns and calculating propagation probabilities. Another line of work touches upon traffic forecasting, with focus on an analysis of spatio-temporal patterns. Yu et al. (2017) made the first attempt to apply deep LSTM recurrent neural network for traffic forecasting and proposed mixture deep LSTM model for post-accident forecasting. Many LSTM-based methods had been used to predict the propagation patterns in the next time steps based on historical data (Basak et al., 2019; Di et al., 2019). Li et al. (2017) proposed a deep learning framework for traffic forecasting, called diffusion convoluional RNN, which incorporates both spatial and temporal dependency in the traffic flow. Li et al. (2021) proposed a multistep traffic forecasting model, the Dynamic Graph Convolutional Network (DGCN). To develop dynamical mappings for spatio-temporal relationships, DGCN adopted a new spatial attention variant considering the upstream–downstream asymmetry of traffic dynamics and the influence of incomplete data. While these approaches relate to time delay estimation in terms of understanding congestion propagation patterns, pattern analysis and traffic forecasting are different from the time delay estimation problem of interest.

Traffic Causality Analysis The traffic causal analysis methods have been developed to identify causal relationships among congested roads and to detect congestion propagation patterns. Bayesian network (BN) approaches were widely implemented to analyse traffic causality. It is a probabilistic graphical model which comprises a set of random variables and their conditional dependencies through directed acyclic graph (Pearl, 1988). To reflect spatial temporal feature of traffic data, Bayesian network was extended into dynamic BN (DBN) which models causality of sequence of variables such as a time series or stochastic process (Ghahramani, 1997). Based on the causality captured via DBN, Sun et al. (2005) forecast future traffic flow by ranking the input variables to identify a subset of BN as the cause nodes using the Pearson correlation coefficient. However, this approach mainly focuses on revealing causal relationship, not providing information about estimated time delay among the roads during congestion being propagated.

Lag Estimation in Multiple Time Series To discover the relationships between neighboring road segments, several methods have been used to analyze interactions among multiple time series, ranging from the detrended cross-correlation analysis (DCCA) cross-correlation coefficient (Zebende, 2011) operating in the time or frequency domains to transfer entory (TE) operating in the information domain. Based on the DCCA cross-correlation coefficient, Shen (2015) proposed a time-lagged DCCA cross-correlation coefficient to quantify the level of time-lagged cross-correlations between two non-stationary time series at different time scales. Chandra & Al-Deek (2008) used cross-correlation analysis to prove significant relationship between the current value of speed at the station in question and the past values of the speed at both upstream and downstream stations in freeway traffic network. In information theory, transfer entropy (TE) (Schreiber, 2000), as a model-free approach, has been a popular measure of the directional interaction between two time series, because of its inherent ability to incorporate directional and dynamical information, its sensitivity to both linear and nonlinear interactions (Barnett et al., 2009). To quantify the information transfer between time series in complex networks and to detect the timing when such a transfer occurs, several lag-specific TE estimators have been proposed (Faes et al., 2014; Wibral et al., 2013). Xiao et al. (2020) used transfer entropy to discover the real dynamic process of flight-delay propagation among multiple airports. However, to the best of our knowledge, our proposed method is the first to use lag-specific TE to predict the time delay for congestion propagation in road traffic networks. The proposed method is compared with the time-lagged DCCA cross-correlation coefficient in Section 5.

3 BACKGROUND

3.1 BOOTSTRAP FOR MARKOV CHAINS

Suppose that $\{X_t\}_{t\geq 1}$ be a stationary Markov chain with a finite state space $S = \{s_1, \ldots, s_n\}$, where $n \in \mathbb{N}$. Let $\mathbf{P} = (p_{ij}) \in \mathbb{R}^{n\times n}$ be a transition probability matrix of the chain and the stationary distribution by $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$. Thus, for any $1 \leq i, j \leq n$, $p_{ij} = P(X_{t+1} = s_j | X_t = s_i)$ and $\pi_i = P(X_t = s_i)$. Given a time series $\{X_1, \ldots, X_L\}$ of size L from a stationary Markov chain, we can estimate π_i and p_{ij} as

$$\hat{\pi}_i = \frac{1}{L}\sum_{t=1}^{L} \mathbb{1}(X_t = s_i), \quad \hat{p}_{ij} = \frac{1}{\hat{\pi}_i L}\sum_{t=1}^{L} \mathbb{1}(X_t = s_i, X_{t+1} = s_j). \tag{1}$$

The bootstrap observations $\{X_1^*, \ldots, X_L^*\}$ can now be generated using the estimated transition matrix and the marginal distribution in Eq.(1) (Kreiss & Lahiri, 2012).

1) Generate a random variable X_1^* from the discrete distribution on $\{1, \ldots, n\}$ that assigns mass $\hat{\pi}_i$ to s_i, $1 \leq i \leq n$.
2) Generate a random variable X_{t+1}^* from the discrete distribution on $\{1, \ldots, n\}$ that assigns mass $\hat{p}_{ij}$ to j, $1 \leq j \leq n$, where s_i is the value of X_t^*.
3) Repeat step 2) until a simulated time series $\{X_1^*, \ldots, X_L^*\}$ has been obtained.

3.2 LAG-SPECIFIC TRANSFER ENTROPY

TE is a measurement of directed information flow (Schreiber, 2000) based on the concept of Shannon entropy (Shannon, 1948) in the area of information theory. For a discrete random variable I with probability distribution $p(i)$, Shannon entropy represents the average number of bits required to optimally encode independent draws, which can be calculated as follows:

$$H(I) = -\sum_i p(i)\log_2 p(i). \tag{2}$$

Eq.(2) can be easily extended to the concept of conditional entropy with two variables. Given two discrete random variables I and J, conditional entropy is defined as

$$H(I|J) = -\sum\sum p(i,j)\log_2 p(i|j) \tag{3}$$

and it can be used to measure the information flow between two discrete random variables.

Consider two discrete random variables, I and J, with marginal probability distributions $p(i)$ and $p(j)$, and joint probability $p(i,j)$. Suppose they are stationary Markov processes of order k and l, respectively. For an order k Markov process I, Eq.(2) can be extended to

$$H^{(k)}(I) = -\sum_{i} p(i_t, i_{t-1}^{(k)}) \log p(i_t | i_{t-1}^{(k)}),$$

where $i_{t-1}^{(k)} = (i_{t-1}, \ldots, i_{t-k})$. Analogously, the information flow from process J to I is measured by quantifying the deviation from the generalized Markov property $p(i_t | i_{t-1}^{(k)}) = p(i_t | i_{t-1}^{(k)}, j_{t-u}^{(l)})$ for an arbitrary source-target lag u, as follows:

$$T_{J\to I}^{(k,l)}(t,u) = \sum p(i_t, i_{t-1}^{(k)}, j_{t-u}^{(l)}) \log \frac{p(i_t | i_{t-1}^{(k)}, j_{t-u}^{(l)})}{p(i_t | i_{t-1}^{(k)})}. \tag{4}$$

Eq.(4) preserves the computational interpretation of TE as an information transfer, which is the only relevant option in keeping with Wiener's principle of causality (Wibral et al., 2013). The transfer entropy is known to be biased in small samples (Marschinski & Kantz, 2002). To correct any bias, (Marschinski & Kantz, 2002) proposed the effective transfer entropy (ETE),

$$ETE_{J\to I}^{(k,l)}(t,u) = T_{J\to I}^{(k,l)}(t,u) - T_{J_{\text{shuffled}}\to I}^{(k,l)}(t,u). \tag{5}$$

where $T_{J_{\text{shuffled}}\to I}^{(k,l)}$ indicates the transfer entropy using a shuffled version of time series J. Shuffling randomly draws values from the original time series J and realigns them to generate a new time series. In this way, shuffling destroys the time series dependencies of J as well as the statistical dependencies between J and I. Note that $T_{J_{\text{shuffled}}\to I}^{(k,l)}$ converges to zero as the sample size increases, and any nonzero value of $T_{J_{\text{shuffled}}\to I}^{(k,l)}(t,u)$ is due to the small sample effects. To derive a consistent estimator, shuffling is repeated, and the average of the resulting shuffled transfer entropy estimates across all replications serves as an estimator for the small sample bias, which is subsequently subtracted from the Shannon or Rényi transfer entropy estimate to obtain a bias-corrected effective transfer entropy estimate.

Using Eq.(5), the time lag in a causal relationship $J \to I$ can be estimated by solving the optimization problem,

$$\hat{u} = \underset{u\in\mathbb{N}}{\operatorname{argmax}}\ ETE_{J\to I}^{(k,l)}(t,u). \tag{6}$$

In this study, we assume $k = \ell = 1$.

4 METHODOLOGY

4.1 PROBLEM DEFINITION

Suppose that information of congestion is transferred from a source road to a destination road with a time delay of u, given a road network. The goal of time delay estimation is to estimate the time delay lag u between the source road and the target road, given previously observed traffic speed data on the two roads. Denote the traffic speed data observed on the source road and the target road as $\{X_t\}_{t=1}^{L}$ and $\{Y_t\}_{t=1}^{L}$, respectively. Then, time delay estimation problem aims to develop a function $f(\cdot)$ that computes the source-target lag u, $\left[\{X_t\}_{t=1}^{L}, \{Y_t\}_{t=1}^{L}\right] \xrightarrow{f(\cdot)} u$.

4.2 DECOMPOSITION AND BOOTSTRAPPING

Consider a time series of congested traffic speed data, $\{X_t\}_{t=1}^{L}$, which have properties of scale-dependence, nonlinearity, and non-stationarity. To identify the causal relationship effectively among such complicated time series, the application of appropriate preprocessing methods is essential.

First of all, we decompose a time series into a trend and its residual as follows:

$$\forall t, X_t = \mathcal{T}_t + R_t = \frac{1}{m}\sum_{j=0}^{m-1} X_{t-j} + R_t, \tag{7}$$

where $\mathcal{T}_t$ is the trend component and R_t is the residual component. The trend component is a one-sided moving average of order m and the average of the forefront value at time t. The purpose of the trend component is to smooth a time series in order to estimate the underlying trend. After extracting the underlying trend from $\{X_t\}_{t=1}^L$, the residual time series $\{R_t\}_{t=1}^L$ is assumed to be a stationary Markov process. The assumption of Markovian property in traffic speed is not new. Many traffic speed prediction or modeling studies have been conducted under this assumption (Hong et al., 2006; Chandra & Al-Deek, 2009; Vlahogianni et al., 2014; Pavlyuk, 2017; Song et al., 2019). Based on $\{R_t\}_{t=1}^L$, we can generate the bootstrap residuals $\{R_t^{*(b)}\}_{t=1}^L$ as explained in Section 3.1. Once the bootstrap residuals are generated, then we easily obtain a bootstrap time series $\{X_t^{*(b)}\}_{t=1}^L$ by $\mathcal{T}_t + R_t^{*(b)}$ for $t = 1, \ldots, L$.

4.3 NORMALIZATION

For the obtained bootstrap time series, we apply nonlinear normalization with a sliding window to handle scale-dependent, nonlinear, and non-stationary time series of traffic speed data. To make them scale-independent and close to linear (Wang et al., 2019), a nonlinear function Φ, which is a standard normal cumulative distribution function, is applied. In addition, a sliding window technique has been applied to handle a non-stationary time series (Ogasawara et al., 2010). Let $\mathbf{X}_{t,w} = \left\{X_k^{*(b)}\right\}_{k=t-w+1}^{t}$ be the forefront sequence of $X_t^{*(b)}$ with a sliding window size of w, and $F_{25,t}$, $F_{50,t}$, and $F_{75,t}$ be the 25th, 50th, and 75th percentiles of $\mathbf{X}_{t,w}$. Note that these percentiles depend on the location of the sliding window. Then, a normalized time series, $\left\{\tilde{X}_t^{*(b)}\right\}_{t=1}^{L}$, can be obtained by

$$\tilde{X}_t^{*(b)} = \Phi\left(0.5 \times \frac{X_t^{*(b)} - F_{50,t}}{F_{75,t} - F_{25,t}}\right). \tag{8}$$

To verify the effectiveness of the nonlinear normalization method in Eq.(8), we compared its performances with existing normalization methods (Wang et al., 2019; Ogasawara et al., 2010), including the min-max method $\left(\tilde{X}_t^{*(b)} = \frac{X_t^{*(b)}}{\max \mathbf{X}_{t,w}}\right)$, and a z-score method $\left(\tilde{X}_t^{*(b)} = \frac{X_t^{*(b)} - \mu(\mathbf{X}_{t,w})}{\sigma(\mathbf{X}_{t,w})}\right)$ with or without a sliding window technique in Section 5.

4.4 SYMBOLIC ENCODING AND TRANSFER ENTROPY

To compute the lag-specific TE in Eq.(4), we discretize continuous data using symbolic encoding. This discretization can be obtained by partitioning the data into a finite number of bins. We denote the bounds specified for the n bins by $q_1, q_2, \ldots, q_{n-1}$, where $q_1 < q_2 < \cdots < q_{n-1}$. For the normalized time series in Eq.(8), we can obtain the encoded time series $\{J_t^{*(b)}\}_{t=1}^L$ by

$$J_t^{*(b)} = \begin{cases} 1 & \text{for } \tilde{X}_t^{*(b)} \leq q_1 \\ 2 & \text{for } q_1 < \tilde{X}_t^{*(b)} < q_2 \\ \vdots & \quad \vdots \\ n & \text{for } \tilde{X}_t^{*(b)} \geq q_{n-1}. \end{cases} \tag{9}$$

The choice of bins is motivated by the distribution of the data. In the case in which tail observations are commonly of particular interest, binning is usually based on empirical quantiles such that the left and right tail observations are selected into separate bins. In this study, we implemented symbolic encoding with $n = 3$ based on 5% and 95% empirical quantiles to focus on the extreme high or low speed caused by dynamic speed change or traffic accidents.

Consequently, we obtain $\{J_t^{*(b)}\}_{t=1}^L$ from $\{\tilde{X}_t^{*(b)}\}_{t=1}^L$ and $\{I_t^{*(b)}\}_{t=1}^L$ from $\{\tilde{Y}_t^{*(b)}\}_{t=1}^L$, respectively, for $b = 1, \ldots, B$. Given $\{J_t^{*(b)}\}_{t=1}^L$ and $\{I_t^{*(b)}\}_{t=1}^L$, $b = 1, \ldots, B$, we obtain bootstrap observations of the time lag, $u^{*(1)}, \ldots, u^{*(B)}$ using Eq.(6) as explained in Section 3.2.

4.5 TIME DELAY ESTIMATION AND HYPERPARAMETER TUNING

Suppose that bootstrap observations of the time lag follows a distribution $\mathcal{G}$,

$$u^{*(1)}, \ldots, u^{*(B)} \sim \mathcal{G},$$

where $\mathcal{G}$ represents the distribution of the time lag, which is unknown in practice. Let μ and σ^2 denote the mean and variance of $\mathcal{G}$, respectively, and they can be estimated by

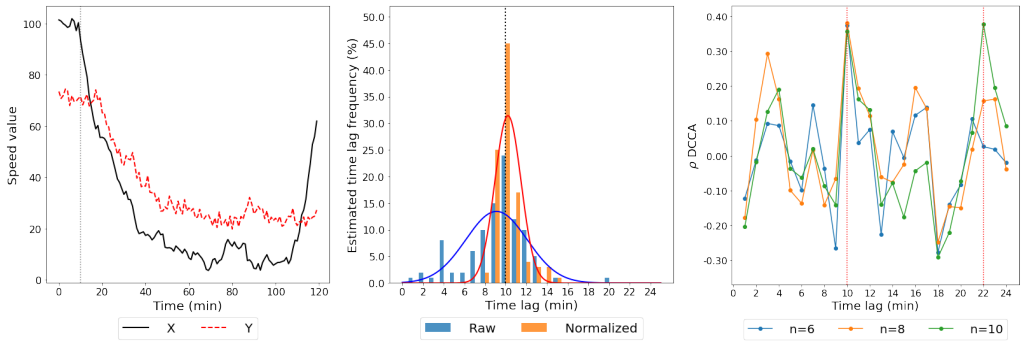
$$\hat{\mu}_B = \frac{1}{B}\sum_{b=1}^{B} u^{*(b)}, \quad \hat{\sigma}_B^2 = \frac{1}{B}\sum_{b=1}^{B}\left(u^{*(b)}\right)^2 - \hat{\mu}_B^2.$$

Proposition 1 implies 1) the bootstrap estimate $\hat{\mu}_B$ is an unbiased estimate of μ, 2) $\frac{1}{B}\hat{\sigma}_B^2$ quantifies the uncertainty of $\hat{\mu}_B$. That is, we can evaluate uncertainty of the bootstrap estimate $\hat{\mu}_B$ using $\frac{1}{B}\hat{\sigma}_B^2$, which is practically useful because μ is unknown. This can also be applied to hyperparameter tuning. In this study, we used a grid search to find a set of hyperparameters (the length of time series (L) and the sliding window size (w)) that minimize $\frac{1}{B}\hat{\sigma}_B^2$.

Proposition 1. *Let $u^{*(1)}, \ldots, u^{*(B)}$ be a bootstrap sample and $E(u^{*(b)}) = \mu$, $Var(u^{*(b)}) = \sigma^2$. Then, sample mean $\hat{\mu}_B = \frac{1}{B}\sum_{b=1}^{B} u^{*(b)}$ approximately follows $\mathcal{N}\left(\mu, \frac{1}{B}\hat{\sigma}_B^2\right)$, where $\hat{\sigma}_B^2$ is the sample variance of the bootstrap sample.*

Proof. Since $\hat{\sigma}_B^2 \to \sigma^2$ in probability, $\frac{\sqrt{B}(\hat{\mu}_B - \mu)}{\hat{\sigma}_B} = \frac{\sigma}{\hat{\sigma}_B}\frac{\sqrt{B}(\hat{\mu}_B - \mu)}{\sigma} \xrightarrow{d} \mathcal{N}(0,1)$ by the Central Limit Theorem and Slutsky's Theorem (Casella & Berger, 2021). Hence, $\hat{\mu}_B$ approximately follows a normal distribution, $\hat{\mu}_B \sim \mathcal{N}\left(\mu, \frac{1}{B}\hat{\sigma}_B^2\right)$. □

5 SIMULATION STUDIES



(a) Simulated data

(b) Proposed method

(c) The time-lagged DCCA cross-correlation coefficient

Figure 2: The results of simulation study

The proposed method was validated using the simulated data. Two time series, $\{X_t\}_{t=1}^{120}$ and $\{Y_t\}_{t=1}^{120}$, are generated by

$$X_t = \begin{cases} 100 + \epsilon_{x,t} & \text{for } t < 10 \\ 0.95X_{t-1} + \epsilon_{x,t} & \text{for } 10 \le t < 95 \\ 1.10X_{t-1} + \epsilon_{x,t} & \text{for } t \le 120 \end{cases}, \quad Y_t = \begin{cases} 70 + \epsilon_{y,t} & \text{for } t < 10 \\ 0.5X_{t-u_0} + 20 + \epsilon_{y,t} & \text{for } t \ge 10 \end{cases},$$

where $\epsilon_{x,t} \sim \mathcal{N}(0,2)$ and $\epsilon_{y,t} \sim \mathcal{N}(0,2)$. A predetermined source-target lag (u_0) exists such that a significant information flow from X to Y is formed, but not vice versa. Figure 2(a) depicts two time series with $u_0 = 10$. The black solid line and red dashed line represent $\{X_t\}_{t=1}^{120}$ and $\{Y_t\}_{t=1}^{120}$, respectively. This simulation represents a typical congestion propagation situation between two adjacent roads, R_X and R_Y, assuming that there was a traffic accident on road R_X at $t = 10$ and that the congestion was resolved at $t = 95$. With the time shift $u_0 = 10$, the congestion on R_X propagates to road R_Y.

The proposed method was applied to the simulated data with $m = 2$ and $w = 20$ as explained in Section 4. Figure 2(b) compares the distributions of bootstrap observations obtained based on two time series without normalization and those obtained based on two time series with nonlinear normalization. The red and blue lines in Figure 2(b) depict the values of $\left(\hat{\mu}_B, \hat{\sigma}_B^2\right)$ in a distribution form, which are $\left(9.54, 3.94^2\right)$ and $\left(10.30, 1.35^2\right)$, respectively. Here, a normal distribution was used for visualization purposes in the figure. As a result, we confirmed that nonlinear normalization with a sliding window improved the accuracy of the time delay estimation, because $1.35^2 < 3.94^2$.

For comparison purposes, the time-lagged DCCA cross-correlation coefficient was also applied with various overlapping box sizes $(6, 8, 10)$ to the simulation data as shown in Figure 2(c). This benchmark method correctly estimated μ_0 for some overlapping box sizes, but it was too sensitive to the changes of overlapping box sizes to be used as time delay estimation. The detailed results with additional 100 simulated datasets are presented in Appendix A.

Table 1: Simulations results comparison ($u_0 = 10$) with $B = 1000$

Decomposition	Normalization	Metrics	Window length 10	20	30	40	120 (all)	Average
false	none	$\hat{\mu}_B$	-	-	-	-	11.23	11.23
		$\hat{\sigma}_B^2$	-	-	-	-	7.03	7.03
		MAE	-	-	-	-	6.13	6.13
	min-max	$\hat{\mu}_B$	12.93	12.64	13.14	12.71	14.89	13.26
		$\hat{\sigma}_B^2$	7.21	7.49	7.27	7.39	6.72	7.21
		MAE	6.73	6.94	6.92	6.82	7.20	6.92
	z-score	$\hat{\mu}_B$	13.06	13.51	12.97	13.29	14.84	13.53
		$\hat{\sigma}_B^2$	6.98	6.97	7.00	7.15	6.73	6.97
		MAE	6.49	6.67	6.52	6.75	7.23	6.73
	nonlinear	$\hat{\mu}_B$	13.29	12.86	12.28	13.17	14.27	13.17
		$\hat{\sigma}_B^2$	7.12	7.24	7.13	7.07	6.89	7.09
		MAE	6.77	6.66	6.36	6.63	7.03	6.69
true	none	$\hat{\mu}_B$	-	-	-	-	9.54	9.54
		$\hat{\sigma}_B^2$	-	-	-	-	3.94	3.94
		MAE	-	-	-	-	2.45	2.45
	min-max	$\hat{\mu}_B$	14.12	10.57	8.88	10.28	16.97	12.16
		$\hat{\sigma}_B^2$	4.24	2.81	2.51	3.53	4.94	3.61
		MAE	4.75	1.88	1.95	2.26	7.38	3.64
	z-score	$\hat{\mu}_B$	10.09	10.28	10.63	11.97	16.83	11.96
		$\hat{\sigma}_B^2$	2.87	3.58	3.90	4.19	5.90	4.09
		MAE	1.58	2.26	2.55	2.75	7.96	3.42
	nonlinear	$\hat{\mu}_B$	10.78	10.30	10.78	10.99	13.38	11.25
		$\hat{\sigma}_B^2$	2.98	1.35	1.49	2.04	6.72	2.91
		MAE	1.53	0.94	1.25	1.72	6.04	2.30

To investigate the performance of the proposed method, we carried out simulation studies under various settings in (1) decomposition, (2) normalization, and (3) the length of sliding window. Their performances were evaluated by $\hat{\mu}_B$, $\hat{\sigma}_B^2$, and MAE, where $MAE = \frac{1}{B}\sum_{i=1}^{B}|\hat{u}_i - u_0|$. The closer the value is to u_0, the better the estimate $\hat{\mu}_B$. The smaller the values of $\hat{\sigma}_B^2$ and MAE are, the better the results. As Table 1 summarized, the decomposition technique improved the overall performance and the nonlinear normalization with $w = 20$ generally performed better than the other normalization methods.

6 Real Data Example : An Accident-driven Traffic Congestion Propagation

The proposed method was applied to discover causal relationships between roads in two real traffic accident cases that occurred in the city of Seoul, South Korea. The blue stars in Figure 3(a) and

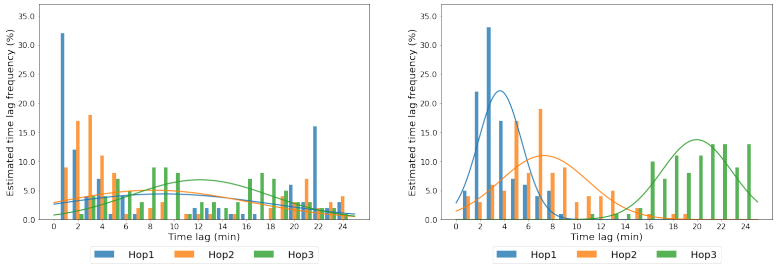
(a) Case 1 (b) Case 2

Figure 3: Two real traffic accident cases that occurred in the city of Seoul

Figure 3(b) depict the exact location where the accident occurred. Case 1 represents a simple road network having a small number of propagation paths involved, whereas case 2 represents a complex road network having many propagation paths involved. Here, we define the propagation path by a sequence of incoming roads in the opposite direction of the traffic flow. For example, case 1 has one propagation path, $[A, B, C, D]$, whereas case 2 has five propagation paths, $[A, B, C, D]$, $[A, E, F, G]$, $[A, H, I, J]$, $[A, H, K, M]$ and $[A, H, K, L]$. Traffic congestion caused by traffic accidents propagates along the propagation paths. The kth element in the propagation path is denoted as Hop(k-1). Hop0 refers to the road where the accident occurred. In this study, we investigated up to $k = 4$. The black, red, blue, and green line segments in the figures indicate Hop0, Hop1, Hop2, and Hop3, respectively.

6.1 Case 1: Simple Traffic Network

The accident occurred on September 8, 2020, at 06:44 AM. TE was calculated using average speed data recorded at one-minute intervals from the previous 1 hour to the next 2 hours based on the time when the accident was reported. Figures 4(a) and 4(b) show the results of time delay estimation for the propagation path $[A, B, C, D]$ for the time series without normalization and the time series with nonlinear normalization, respectively. The time series with nonlinear normalization provided a more consistent estimate of the time delay that increases with each hop. Table 2 confirmed this observation. The values of $\hat{\sigma}_B^2$ with nonlinear normalization are much smaller then the others, which indicates nonlinear normalization makes time delay estimate $\hat{\mu}_B$ to be more reliable. Therefore, we conclude that the congestion effect of the accident propagated along the propagation path to Hop1, Hop2, and Hop3 3.60 min, 7.30 min, and 19.97 min after the accident.



(a) Without normalization (b) Nonlinear normalization

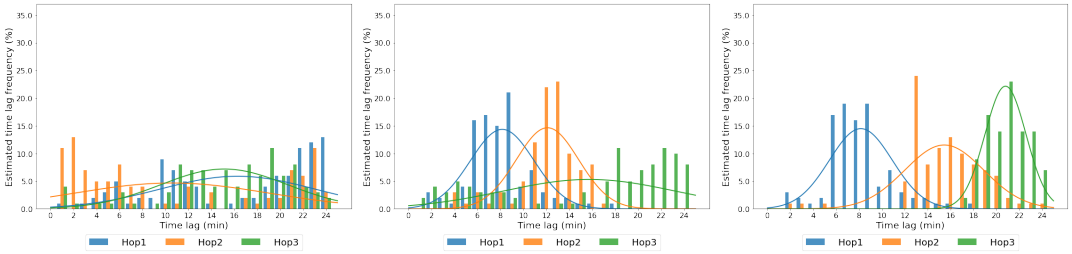
Figure 4: Time delay estimation for Case 1

Table 2: The results of time delay estimation for Case 1

Normalization	**Hop1**		**Hop2**		**Hop3**	
	$\hat{\mu}_B$	$\hat{\sigma}_B^2$	$\hat{\mu}_B$	$\hat{\sigma}_B^2$	$\hat{\mu}_B$	$\hat{\sigma}_B^2$
without normalization	9.62	83.70	8.07	61.31	12.08	34.83
nonlinear normalization	3.60	2.88	7.30	13.83	19.97	8.93

6.2 CASE 2: COMPLEX TRAFFIC NETWORK

The accident occurred on September 4, 2020, at 22:16 PM and affected total five propagation paths. For each path, the previous 1 hour and following 2 hours were considered for time delay estimation. Here, nonlinear normalization was applied and the hyperparameters were tuned by using a grid search. Figure 5 and Table 3 show the results of time delay estimation. In path 1, no specific causal relationship could be found as depicted in Figure 5(a). This finding was supported by the corresponding large values of $\hat{\sigma}_B^2$ in the table. Similarly, we conclude that the congestion effect of the accident at road A did not propagate along the path 2, but propagated along the paths 3, 4 and 5. Moreover, the values of $\hat{\sigma}_B^2$ in Table 3 indicate that the congestion effect propagated along the path 3 up to Hop2 as depicted in Figure 5(b), and along paths 4 and 5 up to Hop3 as depicted in Figure 5(c). For paths 4 and 5, the time delay estimates are (8.23, 15.65, 22.06) and (8.22, 15.55, 20.75), respectively.



(a) Path 1 (b) Path 3 (c) Path 5

Figure 5: Time delay estimation for Case 2

Table 3: The results of time delay estimation for Case 2

Propagation path	**Hop1**		**Hop2**		**Hop3**	
	$\hat{\mu}_B$	$\hat{\sigma}_B^2$	$\hat{\mu}_B$	$\hat{\sigma}_B^2$	$\hat{\mu}_B$	$\hat{\sigma}_B^2$
Path 1: $[A, B, C, D]$	16.03	47.09	11.59	73.56	15.17	30.46
Path 2: $[A, E, F, G]$	4.93	19.19	7.20	21.58	7.04	14.54
Path 3: $[A, H, I, J]$	8.21	7.97	12.13	7.03	15.72	57.16
Path 4: $[A, H, K, M]$	8.23	7.62	15.65	9.01	22.06	3.24
Path 5: $[A, H, K, L]$	8.22	7.67	15.55	11.01	20.75	3.17

7 CONCLUSION

Traffic congestion due to an accident spreads its effects to the inflow roads, which creates a causal relationship between the accident road and the inflow roads. To identify the causal relationship between them, we have proposed a new method for estimating the differences in congestion time. The proposed method utilizes a lag-specific TE estimator with decomposition and normalization techniques. This paper conducted extensive performance comparisons with various experimental settings and found that the proposed decomposition and nonlinear normalization methods outperform those without them. Moreover, the bootstrap technique and its density estimation of statistical functionals enable uncertainty quantification of the time delay estimates. This uncertainty quantification allows us to evaluate the reliability of time delay estimates and serves as a basis for optimal hyperparameter tuning. It is also confirmed that the proposed method provided more stable and robust results than existing time lag estimation methods such as time-lagged DCCA cross-correlation coefficient. The proposed time delay estimation helps to quantitatively understand the propagation of traffic congestion and can be used in GPS navigation systems to provide users with a more accurate ETA. However, there are limitations that need to be addressed. In the proposed method, $\hat{\sigma}_B^2$ serves as a key indicator of a causal relationship between two time series, but lacks guidance on how to make a decision based on its value. The rigorous and practical decision-making based on $\hat{\sigma}_B^2$ will be left for future research.

REPRODUCIBILITY STATEMENT

We implemented simulation studies in Python 3.8.0 on AMD Ryzen Threadripper 3970X 32-Core processor workstation. For calculating the effective transfer entropy (ETE), we relied on the R package, `RTransferEntropy` (version 0.2.13). All scripts for simulation studies are available on `https://bit.ly/3Bczn1n`.

REFERENCES

Lionel Barnett, Adam B Barrett, and Anil K Seth. Granger causality and transfer entropy are equivalent for gaussian variables. *Physical review letters*, 103(23):238701, 2009.

Sanchita Basak, Abhishek Dubey, and Leao Bruno. Analyzing the cascading effect of traffic congestion using lstm networks. In *2019 IEEE International Conference on Big Data (Big Data)*, pp. 2144–2153. IEEE, 2019.

Stefano Bianchi. fathon: A python package for a fast computation of detrendend fluctuation analysis and related algorithms. *Journal of Open Source Software*, 5(45):1828, 2020.

George Casella and Roger L Berger. *Statistical inference*. Cengage Learning, 2021.

Srinivasa Ravi Chandra and Haitham Al-Deek. Cross-correlation analysis and multivariate prediction of spatial time series of freeway traffic speeds. *Transportation Research Record*, 2061(1): 64–76, 2008.

Srinivasa Ravi Chandra and Haitham Al-Deek. Predictions of freeway traffic speeds and volumes using vector autoregressive models. *Journal of Intelligent Transportation Systems*, 13(2):53–72, 2009.

Petros Damos. Using multivariate cross correlations, granger causality and graphical models to quantify spatiotemporal synchronization and causality between pest populations. *BMC ecology*, 16(1):1–17, 2016.

Xiaolei Di, Yu Xiao, Chao Zhu, Yang Deng, Qinpei Zhao, and Weixiong Rao. Traffic congestion prediction by spatiotemporal propagation patterns. In *2019 20th IEEE International Conference on Mobile Data Management (MDM)*, pp. 298–303. IEEE, 2019.

Luca Faes, Daniele Marinazzo, Alessandro Montalto, and Giandomenico Nollo. Lag-specific transfer entropy as a tool to assess cardiovascular and cardiorespiratory information transfer. *IEEE Transactions on Biomedical Engineering*, 61(10):2556–2568, 2014.

Zoubin Ghahramani. Learning dynamic bayesian networks. In *International School on Neural Networks, Initiated by IIASS and EMFCSC*, pp. 168–197. Springer, 1997.

Wei-Chiang Hong, Ping-Feng Pai, Shun-Lin Yang, and Robert Theng. Highway traffic forecasting by support vector regression model with tabu search algorithms. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pp. 1617–1621. IEEE, 2006.

Jens-Peter Kreiss and Soumendra Nath Lahiri. Bootstrap methods for time series. In *Handbook of statistics*, volume 30, pp. 3–26. Elsevier, 2012.

Guopeng Li, Victor L Knoop, and Hans van Lint. Multistep traffic forecasting by dynamic graph convolution: Interpretations of real-time spatial correlations. *Transportation Research Part C: Emerging Technologies*, 128:103185, 2021.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

Wei Liu, Yu Zheng, Sanjay Chawla, Jing Yuan, and Xie Xing. Discovering spatio-temporal causal interactions in traffic data streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1010–1018, 2011.

Robert Marschinski and Holger Kantz. Analysing the information flow between financial time series. *The European Physical Journal B-Condensed Matter and Complex Systems*, 30(2):275–281, 2002.

Hoang Nguyen, Wei Liu, and Fang Chen. Discovering congestion propagation patterns in spatio-temporal traffic data. *IEEE Transactions on Big Data*, 3(2):169–180, 2016.

Eduardo Ogasawara, Leonardo C Martinez, Daniel De Oliveira, Geraldo Zimbrão, Gisele L Pappa, and Marta Mattoso. Adaptive normalization: A novel data normalization approach for non-stationary time series. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2010.

Julian D Olden and Bryan D Neff. Cross-correlation bias in lag analysis of aquatic time series. *Marine Biology*, 138(5):1063–1070, 2001.

Dmitry Pavlyuk. Short-term traffic forecasting using multivariate autoregressive models. *Procedia Engineering*, 178:57–66, 2017.

J Pearl. Morgan kaufmann series in representation and reasoning. *Probabilistic reasoning in intelligent systems: Networks of plausible inference. San Mateo, CA, US: Morgan Kaufmann*, 1988.

Boris Podobnik and H Eugene Stanley. Detrended cross-correlation analysis: a new method for analyzing two nonstationary time series. *Physical review letters*, 100(8):084102, 2008.

Wolfgang Nikolaus Probst, Vanessa Stelzenmüller, and Heino Ove Fock. Using cross-correlations to assess the relationship between time-lagged pressure and state indicators: an exemplary analysis of north sea fish population indicators. *ICES journal of marine science*, 69(4):670–681, 2012.

Thomas Schreiber. Measuring information transfer. *Physical review letters*, 85(2):461, 2000.

Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Chenhua Shen. Analysis of detrended time-lagged cross-correlation between two nonstationary time series. *Physics Letters A*, 379(7):680–687, 2015.

Zhanguo Song, Yanyong Guo, Yao Wu, and Jing Ma. Short-term traffic speed prediction under different data collection time intervals using a sarima-sdgm hybrid prediction model. *PloS one*, 14(6):e0218626, 2019.

Shiliang Sun, Changshui Zhang, and Yi Zhang. Traffic flow forecasting using a spatio-temporal bayesian network predictor. In *International conference on artificial neural networks*, pp. 273–278. Springer, 2005.

Eleni I Vlahogianni, Matthew G Karlaftis, and John C Golias. Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, 43: 3–19, 2014.

Jingyu Wang, Sheng Su, Yinhong Li, Jinfu Chen, and Dongyuan Shi. Desaturated probability integral transform for normalizing power system measurements in data-driven manipulation detection. In *2019 IEEE Power & Energy Society General Meeting (PESGM)*, pp. 1–5. IEEE, 2019.

Michael Wibral, Nicolae Pampu, Viola Priesemann, Felix Siebenhühner, Hannes Seiwert, Michael Lindner, Joseph T Lizier, and Raul Vicente. Measuring information-transfer delays. *PloS one*, 8 (2):e55809, 2013.

Yinhong Xiao, Yaoshuai Zhao, Ge Wu, and Yizhen Jing. Study on delay propagation relations among airports based on transfer entropy. *IEEE Access*, 8:97103–97113, 2020.

Haoyi Xiong, Amin Vahedian, Xun Zhou, Yanhua Li, and Jun Luo. Predicting traffic congestion propagation patterns: a propagation graph approach. In *Proceedings of the 11th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, pp. 60–69, 2018.

Rose Yu, Yaguang Li, Cyrus Shahabi, Ugur Demiryurek, and Yan Liu. Deep learning: A generic approach for extreme condition traffic forecasting. In *Proceedings of the 2017 SIAM international Conference on Data Mining*, pp. 777–785. SIAM, 2017.

Gillney Figueira Zebende. Dcca cross-correlation coefficient: Quantifying level of cross-correlation. *Physica A: Statistical Mechanics and its Applications*, 390(4):614–618, 2011.

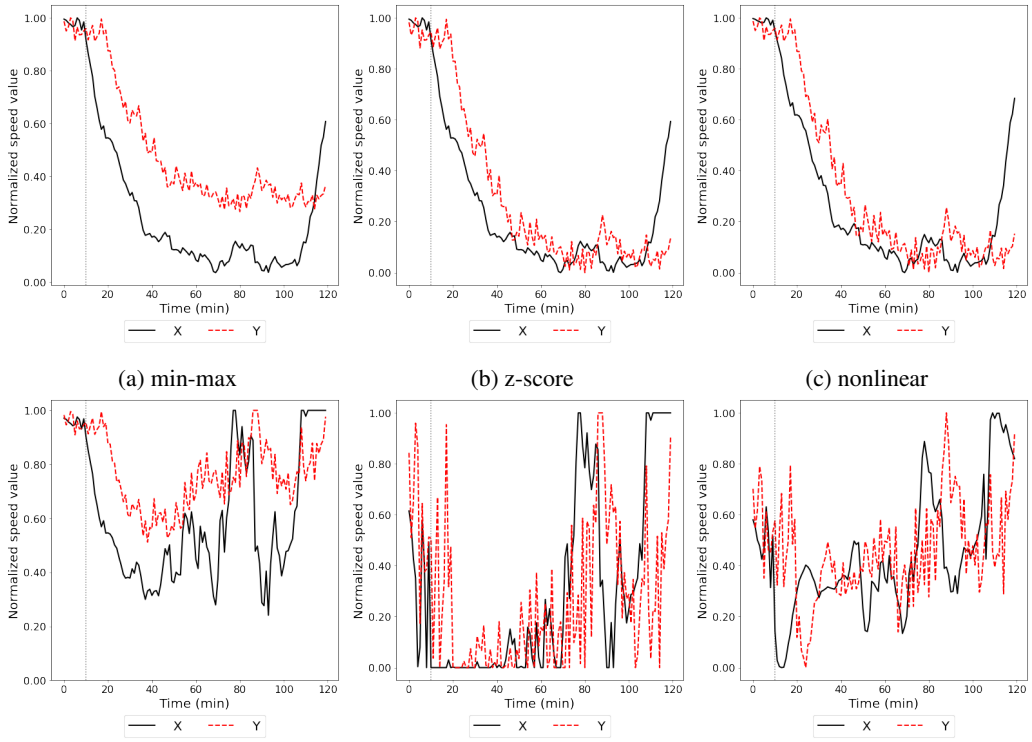
A Detailed Results of Simulation Studies

A.1 Experiment Environment

We implemented simulation studies in Python 3.8.0 on AMD Ryzen Threadripper 3970X 32-Core processor workstation. For calculating the effective transfer entropy (ETE), we relied on the R package, `RTransferEntropy` (version 0.2.13) [1]. All scripts for simulation studies are available on `https://bit.ly/3Bczn1n`.

A.2 Supplementary Results for Table 1 in the Manuscript

We applied three different normalization methods with or without sliding window. Figure 6 shows their results for the time series X and Y, respectively. The top row of Figure 6 represents time series after applying normalization methods without sliding window, while the bottom row illustrates time series after applying normalization methods with sliding window ($w = 20$). The histograms in Figure 7 show the bootstrap distribution for 100 resamples, $\mu^{*(b)}$, $b = 1, \ldots, 100$, for three different normalization methods with or without sliding window. Figure 6(f) and Figure 7(f) confirmed that the nonlinear normalization with sliding window outperforms the others, because it better reflects the local volatility.



(a) min-max (b) z-score (c) nonlinear

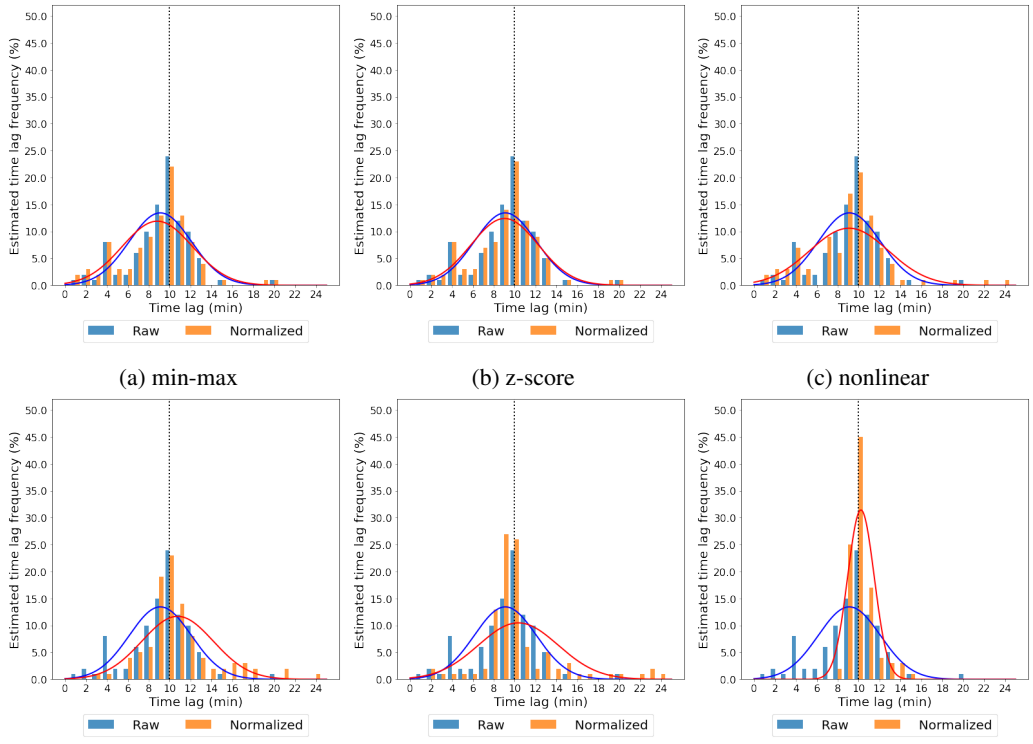
(d) min-max ($w = 20$) (e) z-score ($w = 20$) (f) nonlinear ($w = 20$)

Figure 6: Comparison of normalization methods

A.3 Supplementary Results for Hyperparameter Tuning

For hyperparameter tuning in the time delay estimation, we used a grid search to find a set of hyperparameters that minimize only $\frac{1}{B}\hat{\sigma}^2_B$, because true μ (μ_0) is unknown in reality. To justify the objective function in hyperparameter optimization, we showed that the bootstrap estimate ($\hat{\mu}_B$) approaches μ_0 as $\frac{1}{B}\hat{\sigma}^2_B$ decreases. To this end, we investigated the relationship between

[1] https://github.com/BZPaper/RTransferEntropy



(a) min-max (b) z-score (c) nonlinear

(d) min-max ($w = 20$) (e) z-score ($w = 20$) (f) nonlinear ($w = 20$)

Figure 7: Comparison of Bootstrap estimation results

Diff ($= |\hat{\mu}_B - u_0|$) and $\hat{\sigma}_B^2$ using further simulation studies for three different normalization methods under various experimental settings:

$$\mu_0 = \{5, 10, 15\}, \quad \sigma_x = \sigma_y = \{1, 2, 3\}, \quad w = \{10, 20, 30, 40\}.$$

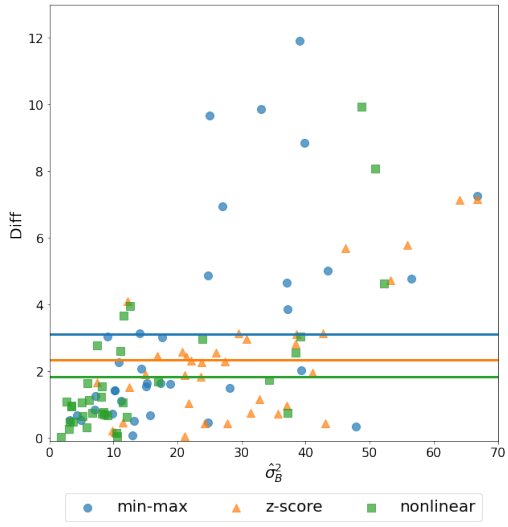


Figure 8: Relationship between *Diff* and $\hat{\sigma}_B^2$

Figure 8 shows a scatter plot whose x-axis and y-axis represent $\hat{\sigma}_B^2$, *Diff*, respectively. The shapes of points indicate the type of normalization method and the horizontal lines are the average *Diff*

values of the points corresponding to the normalization method. As we expected, we observed that $|\hat{\mu}_B - u_0|$ tends to approach 0 as $\hat{\sigma}_B^2$ decreases. Moreover, we confirmed that the nonlinear normalization method is generally superior to the other methods. Tables 4 and 5 present detailed simulation results used to construct the scatter plot. Each number in the table represent the mean value of 100 random simulations with different lags and noise levels. Bold numbers in the tables represent the smallest numbers within the same μ_0 and noise level.

Table 4: Detailed results of simulation studies on *Diff*

Methods	**Window length**	**Experiment settings** (u_0 and noise level)								
		5			**10**			**15**		
		1	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
min-max	**10**	9.86	6.94	11.92	9.66	4.66	5.02	1.63	2.02	0.33
	20	7.25	8.84	1.50	3.85	4.87	0.52	0.72	0.46	2.28
	30	0.54	4.78	0.86	0.69	0.69	1.12	1.43	**0.07**	3.15
	40	1.42	1.64	1.26	3.01	3.05	0.54	1.56	1.64	2.07
	120 (all)	2.04	5.40	0.77	0.87	3.37	1.88	1.31	0.60	0.06
z-score	**10**	3.13	4.73	5.70	1.85	2.83	0.97	0.45	0.45	1.96
	20	2.96	5.79	2.57	2.27	3.12	0.75	0.73	0.44	2.29
	30	3.15	7.17	1.88	1.53	4.10	0.47	0.21	2.46	1.04
	40	1.66	7.14	2.32	2.45	1.91	0.18	0.06	1.16	2.58
	120 (all)	2.04	5.38	0.78	0.86	3.37	1.88	1.30	0.59	**0.05**
nonlinear	**10**	3.04	8.08	9.94	0.63	4.63	1.06	**0.04**	2.56	0.75
	20	**0.02**	2.96	1.75	0.69	3.67	0.96	0.32	0.79	1.69
	30	0.68	1.14	1.07	0.72	2.79	1.09	0.75	1.24	2.60
	40	0.27	**0.94**	0.65	**0.49**	**0.50**	**0.15**	1.54	1.64	3.95
	120 (all)	0.79	4.06	**0.41**	0.85	3.32	1.78	0.55	0.89	0.64

Table 5: Detailed results of simulation studies on $\hat{\sigma}_B^2$

Methods	**Window Length**	**Experiment settings** (u_0 and noise level)								
		5			**10**			**15**		
		1	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
min-max	**10**	33.03	27.05	39.09	25.01	37.11	43.50	18.87	39.30	47.86
	20	66.79	39.80	28.20	37.18	24.76	13.19	9.89	24.72	10.85
	30	4.96	56.50	7.03	4.33	15.74	11.22	10.17	12.91	14.13
	40	10.23	15.20	7.16	17.59	9.07	3.21	15.10	17.33	14.29
	120 (all)	38.49	52.71	6.90	6.24	40.73	11.54	8.35	**4.59**	9.41
z-score	**10**	42.67	53.19	46.17	23.69	38.40	37.03	24.21	43.07	41.04
	20	30.77	55.88	26.03	23.75	38.58	31.42	35.65	27.73	27.36
	30	29.56	66.75	21.16	12.43	12.23	11.47	9.86	16.86	21.80
	40	7.44	63.99	22.09	21.44	15.01	10.17	21.09	32.85	20.70
	120 (all)	38.48	52.61	6.96	6.24	40.79	11.54	8.33	4.61	9.44
nonlinear	**10**	39.19	50.86	48.71	12.15	52.29	11.45	10.57	38.39	37.21
	20	**1.82**	23.86	34.33	8.56	11.53	3.48	5.77	8.47	16.96
	30	9.14	6.18	**5.13**	8.19	7.44	**2.64**	6.64	8.02	11.08
	40	3.07	**3.40**	5.24	**3.15**	**3.64**	10.44	8.24	5.90	12.54
	120 (all)	25.15	47.38	8.06	7.34	42.60	10.43	**5.46**	5.78	**7.63**

A.4 Supplementary Results for Performance comparison with Baseline

We compare the performance of the proposed method with time lagged correlation with Detrended cross correlation analysis (DCCA). We compare the MAE score of estimated time lag with a different time lag and noise level with 100 different simulation samples. The numbers in the Table 6 represents the mean value of all experiments.

Time lagged correlation with Pearson's correlation coefficient is a typical approach to analyze the relationship between time series. This approach is helpful for identifying lags of the X_t that might be

useful predictors of Y_t. When there is time lag of u between X_t and Y_t, X_t and shifted time series Y_{t+u} would have higher correlation coefficient (Olden & Neff, 2001; Chandra & Al-Deek, 2008; Probst et al., 2012; Damos, 2016). However, Pearson's correlation should be applied to stationary time series. Thus we use DCCA coefficient instead of the Pearson's correlation coefficient. DCCA can estimate lag between time series using detrended fluctuation analysis (Podobnik & Stanley, 2008; Zebende, 2011; Shen, 2015; Bianchi, 2020). Shen (2015) proposed time lagged DCCA, which is similar concept of TLCC.

Here we use DCCA coefficient with a various overlapping box size $n(= 6, 8, 10)$. The performance is sensitive to hyperparameter n and it is not consistent with different simulation settings. Also, it shows larger error when the noise level is high. Thus it shows unstable results for lag estimation Compare to time lagged DCCA coefficient, the proposed approach is much stable in the perspective of MAE value. Furthermore, the proposed method can provide distribution of estimated time lag rather than fixed value as we explained in the manuscript.

Table 6: Performance comparison of proposed approach on MAE

Methods	**Window Length**	**Experiment settings** (u_0 and noise level)								
		5			**10**			**15**		
		1	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
DCCA(6)	-	0.12	16.83	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	7.00	7.00
DCCA(8)	-	0.12	**0.00**	2.50	**0.00**	**0.00**	**0.00**	**0.00**	7.00	7.00
DCCA(10)	-	0.12	16.83	**0.00**	**0.00**	12.00	12.00	**0.00**	**0.00**	**0.00**
without normalization	-	3.34	6.19	1.54	1.49	4.85	2.70	1.82	1.48	2.03
min-max	**10**	10.14	7.51	12.07	10.05	6.49	6.79	3.43	5.86	5.68
	20	7.82	9.18	3.64	4.99	5.81	2.81	1.96	3.96	3.33
	30	1.76	5.85	2.35	1.66	2.97	2.63	2.34	2.66	3.92
	40	2.07	2.72	2.30	3.77	3.32	1.43	3.08	3.39	3.78
	120 (all)	3.33	6.20	**1.53**	1.49	4.85	2.70	1.83	**1.47**	2.02
z-score	**10**	4.22	6.12	6.72	3.35	5.34	5.11	3.05	5.54	5.31
	20	3.39	6.98	4.00	3.57	5.80	4.35	4.16	4.04	4.47
	30	3.69	8.05	3.68	2.23	4.56	2.44	1.57	3.93	3.86
	40	1.79	7.68	3.78	3.36	3.50	2.37	2.71	4.79	4.11
	120 (all)	3.33	6.18	1.54	1.50	4.86	2.70	1.82	1.48	2.02
nonlinear	**10**	3.49	8.69	10.03	2.09	7.29	1.61	2.03	5.52	4.32
	20	**0.69**	3.52	3.53	1.69	4.00	**1.12**	1.42	2.02	3.34
	30	1.46	1.82	1.78	1.73	2.98	1.50	1.61	2.33	3.46
	40	1.37	**1.73**	2.02	**1.29**	**1.46**	2.47	2.32	2.19	4.23
	120 (all)	2.35	5.05	1.65	1.80	5.00	2.61	**1.30**	1.81	**1.92**

B Detailed Results of Real Data Example

B.1 Case 2: Complex Traffic Network

Accident occurred at the 2nd lane out of 5 lanes of the expressway on September 4th 22:16. Previous 1 hour and following 2 hours are considered to calculate transfer entropy. Case 2 has five propagation paths, $[A, B, C, D]$, $[A, E, F, G]$, $[A, H, I, J]$, $[A, H, K, M]$ and $[A, H, K, L]$. Some paths are directly affected by root road and other are not affected. Suggested approach can investigate which road has causal relationship with accident road.

Due to space limitations in the manuscript, we could not show results for all five paths. Tables 7 presents detailed results of real data example used to construct Figure 9. Figure 10 presents the effective transfer entropy (ETE), which is Eq. (6) in the manuscript, and the bootstrap distribution over $u \in \mathbb{N}$ for all five paths.

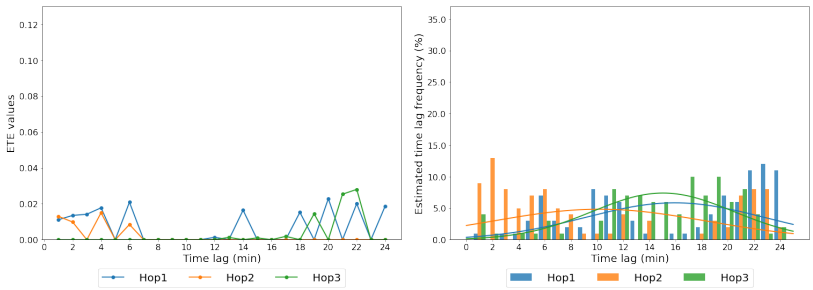
In hyperparameter optimization, we need to decide the sliding window size w and the sequence length L, which affect the accuracy of the time delay estimate. To find the optimal (w, L) values, we conducted the grid search over hyperparameter space. In the following table, The three numbers in square brackets mean the value of $\hat{\sigma}_B^2$ for Hop1, Hop2, and Hop3, respectively.

- Path 1: $[A, B, C, D]$
- Path 2: $[A, E, F, G]$
- Path 3: $[A, H, I, J]$
- Path 4: $[A, H, K, M]$
- Path 5: $[A, H, K, L]$

Figure 9: Traffic network for Case 2

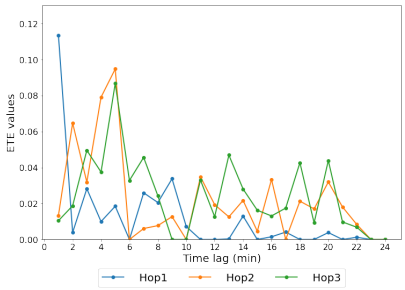
Table 7: Detailed results of real data example on $\hat{\sigma}_B^2$

Path #	**Window length**	$L = 120$	180	240
Path 1	**20**	[10.37, 15.18, 6.35]	[56.10, 9.44, 1.03]	[41.09, 9.64, 0.93]
	40	[2.86, 8.08, 8.83]	[3.31, 6.91, 11.06]	[6.65, 15.14, 26.83]
	60	[3.52, 19.75, 8.03]	[3.04, 13.35, 8.57]	[2.05, 11.71, 1.77]
	80	[15.16, 37.17, 27.01]	[3.89, 21.37, 17.19]	[5.69, 8.56, 10.38]
Path 2	**20**	[1.45, 13.55, 8.25]	[2.81, 16.66, 11.12]	[1.46, 17.41, 10.39]
	40	[2.71, 4.29, 4.31]	[2.07, 3.23, 3.15]	[2.59, 8.72, 5.09]
	60	[8.58, 10.68, 7.66]	[3.81, 6.95, 16.16]	[4.58, 2.48, 7.93]
	80	[13.45, 28.39, 12.03]	[5.98, 14.72, 15.9]	[6.24, 7.63, 12.54]
Path 3	**20**	[2.15, 17.79, 21.82]	[19.15, 26.75, 14.82]	[14.55, 25.88, 14.95]
	40	[3.91, 8.69, 9.77]	[2.88, 7.02, 3.36]	[2.67, 5.03, 4.64]
	60	[17.41, 12.29, 57.25]	[7.73, 7.44, 57.58]	[2.56, 4.49, 40.95]
	80	[10.95, 9.56, 7.7]	[5.79, 4.46, 7.74]	[4.72, 10.72, 9.69]
Path 4	**20**	[2.27, 25.78, 2.11]	[18.23, 25.19, 9.86]	[14.42, 18.74, 15.05]
	40	[3.91, 22.42, 1.31]	[2.61, 27.66, 0.65]	[2.67, 61.19, 2.54]
	60	[17.96, 31.64, 4.89]	[7.67, 11.1, 3.22]	[3.59, 17.43, 1.3]
	80	[10.23, 29.86, 7.79]	[5.94, 14.54, 2.95]	[6.34, 11.82, 5.75]
Path 5	**20**	[2.28, 26.26, 29.79]	[19.15, 25.13, 54.45]	[14.1, 19.33, 58.18]
	40	[3.9, 22.14, 3.53]	[3.33, 27.98, 1.51]	[2.77, 60.14, 0.92]
	60	[19.45, 30.48, 4.94]	[7.67, 10.57, 3.95]	[3.52, 18.62, 5.67]
	80	[10.63, 30.37, 29.91]	[5.88, 14.19, 7.66]	[4.03, 11.93, 6.13]

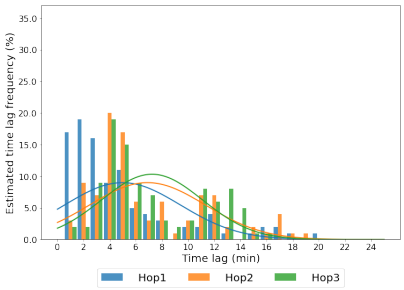


(a) Path 1: ETE

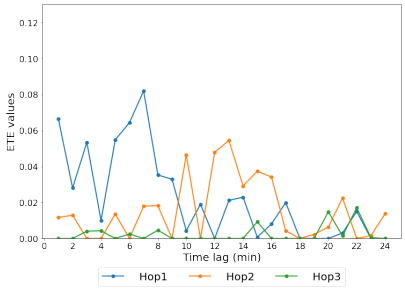
(b) Path 1: bootstrap distribution



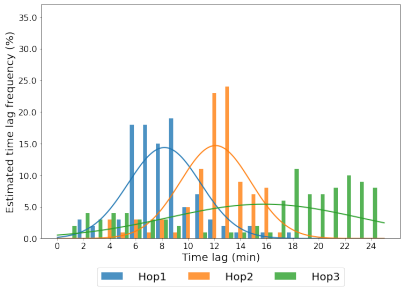
(c) Path 2: ETE



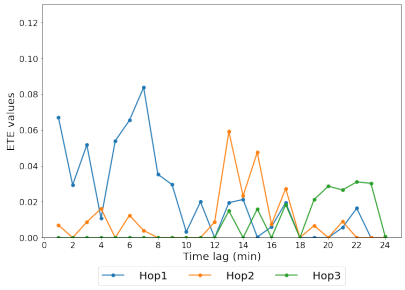
(d) Path 2: bootstrap distribution



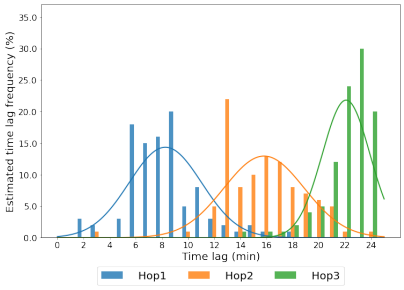
(e) Path 3: ETE



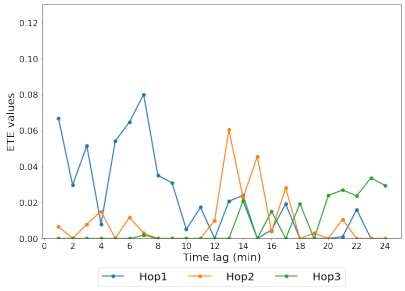
(f) Path 3: bootstrap distribution



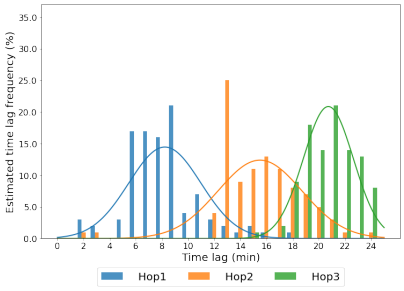
(g) Path 4: ETE



(h) Path 4: bootstrap distribution



(i) Path 5: ETE



(j) Path 5: bootstrap distribution

Figure 10: Case 2. ETE and bootstrap distribution for all five paths