The following publication Divya Saxena and Jiannong Cao. 2022. Multimodal Spatio-Temporal Prediction with Stochastic Adversarial Networks. ACM Trans. Intell. Syst. Technol. 13, 2, Article 18 (April 2022), 23 pages is available at https://dx.doi.org/10.1145/3458025.

Multimodal Spatio-Temporal Prediction with Stochastic Adversarial Networks

DIVYA SAXENA

Department of Computing, The Hong Kong Polytechnic University, Hong Kong, divsaxen@comp.polyu.edu.hk

JIANNONG CAO

Department of Computing and UBDA, The Hong Kong Polytechnic University, Hong Kong, csjcao@comp.polyu.edu.hk

Spatio-temporal (ST) data is a collection of multiple time series data with different spatial locations and is inherently stochastic and unpredictable. An accurate prediction over such data is an important building block for several urban applications, such as taxi demand prediction, traffic flow prediction, etc. Existing deep learning based approaches assume that outcome is deterministic and there is only one plausible future; therefore, cannot capture the multimodal nature of future contents and dynamics. In addition, existing approaches learn spatial and temporal data separately as they assume weak correlation between them. To handle these issues, in this paper, we propose a stochastic spatio-temporal generative model (named D-GAN) which adopts Generative Adversarial Networks (GANs)-based structure for more accurate ST prediction in multiple time steps. D-GAN consists of two components: 1) spatio-temporal correlation network which models spatio-temporal joint distribution of pixels and supports a stochastic sampling of latent variables for multiple plausible futures; 2) a stochastic adversarial network to jointly learn generation and variational inference of data through implicit distribution modelling. D-GAN also supports fusion of external factors through explicit objective to improve the model learning. Extensive experiments performed on two real-world datasets show that D-GAN achieves significant improvements and outperforms baseline models.

CCS Concepts: • **Information systems** → **Location based services**; **Data stream mining**;

Additional Key Words and Phrases: Generative adversarial networks, Spatio-temporal prediction, Deep learning

1 INTRODUCTION

With the rapid development in computer vision and artificial intelligence, a multitude of important research problems on spatio-temporal (ST) predictive learning have emerged and attracted much interest in the research communities [1]–[10]. It has been well-studied in last few years due to their enormous prospect for several real-world applications, such as traffic/crowd flows prediction [5]–[9], [11], [12], precipitation forecasting [13], air and water quality forecasting [14], cellular traffic prediction [4], and demand prediction [3]. While, ST predictive learning for such applications is challenging due to the complex spatial dependencies and temporal dynamics. A ST prediction model must be able to capture ST dynamics between the previous spatio-temporal sequence and future frames accurately. However, capturing these dynamics among the high-dimensional ST data is non-trivial due to the diverse and different types of stochastic events that can occur in ST data.

Recent studies have shown remarkable success for deep learning based spatio-temporal prediction as discussed in [15] . In most of the works, the whole area of a city is partitioned into a grid map (i.e., an image) based on the latitude and longitude. To model the spatial and temporal structure on these grid maps, a combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been widely used. [16] used the CNN-based method (DeepST) to model the spatial and temporal features, while [2] used the residual structures for the same purpose. In [2] and [17], different network of residual CNNs are used to model the spatial

data mapped from different time periods (hourly, daily, and weekly). Similarly, in [11] and [18], the spatial information is first modelled by CNNs or GCNs and then result is passed to the LSTM to get the final output. However, there are a number of challenges limiting the performance of ST prediction in complex scenarios.

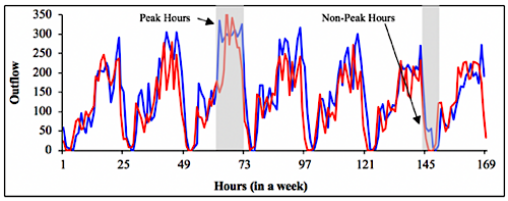

Figure. 1. High variations in the ST data volume, e.g., taxi demand variation pattern for two consecutive weeks in a PoI area of NYC

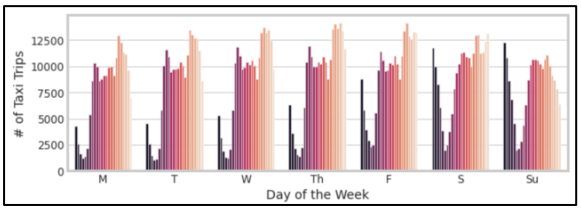

Figure 2(a). Distribution of number of Taxi trips in NYC for each hour in a day of the week

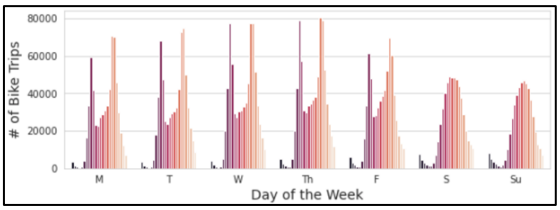

Figure 2(b). Distribution of number of Bike trips in NYC for each hour in a day of the week

***Spatio-temporal correlations*.** To model dynamic relationships between spatial and temporal features simultaneously and finding their impact on the data variations is a non-trivial problem. Figure 1 shows a scenario of ST correlations in which for a PoI area of NYC, taxi demand is high from the evening to night while taxi demand is low in the early morning. This scenario also shows the high variations in the ST data as it is inherently dynamic and changing over time (i.e., stochastic). Through Fig. 1, it can be observed that both spatial attributes (location, i.e., PoI area) and temporal attributes (i.e., time) have high influence on the variations of the ST data (i.e., outflow trend). In addition, several latent events, such as blocked driveway, road construction, traffic jam, etc., happen suddenly and over a very short period of time in a location cannot be observed through data, but have hidden influence on the variations of the ST data. Also, the frequency of such events depends on both spatial and temporal attributes, i.e., some locations are more prone to traffic jam at particular period of time. Existing ST prediction methods [2][11][17][18], have separate models for each modality and fuse the learned spatial and temporal features in the final layer for the prediction. Modelling spatial and temporal dependencies independently shows that existing models assume that the relationships between these dependencies are weak; therefore, cannot model data variations accurately. Thus, *how to jointly learn spatial dependencies at each timestamp and temporal evolution is a major challenge*.

Multiple types of temporal correlations. Figure 2 (a) and (b) show the trips distribution for each hour of the week for Taxi and Bike datasets, respectively. Taxi and Bike trip distribution is showing that taxi and bike demand values are highly influenced by hour of a day, day in a week and hour in a week. Existing approaches use the one level data mapping (i.e., hour/day/week) as an input to the model which can lead to low accuracy due to the lack of modelling complicated temporal dynamics in the ST data of real-world. Thus, *how to decide input to the model to capture long and complicated temporal dynamics in the real-world spatio-temporal prediction problem is a challenging task*.

Multimodality. In the case of high-dimensional data and the complex dynamics of the environment, modelling multimodality in prediction is a hard problem [19]. Existing spatio-temporal prediction approaches have assumed that the environment is deterministic and there is only one plausible future. Due to this assumption, prediction model leads to poor performance in the case of handling stochastic dynamics in real-world environment. Therefore, when a model trained using the mean squared error (MSE) *by construction* between the ground truth and the predicted data, model chooses an average over many slightly different data. This causes the low accuracy in the long-term prediction [20]. When a model trained using both regression and adversarial loss, model performs better than models with means, but produces only single mode of the distribution instead of having multimodal training dataset, as pointed out in [19]. Therefore, we can say multimodality is extremely difficult to handle by standard regressors or classifiers as standard regressors based models choose a mean and classifiers based models only produce a good fit to the principal mode and could not generate quality data. Thus, *how to handle such multimodality is quite challenging*.

External Factors Fusion. Spatio-temporal prediction is enormously a challenging task due to the inherent uncertainty of the future and numerous factors of the variation in ST data causing complex dynamics in raw pixel values. ST prediction is highly influenced by many external factors, such as time factors, adverse weather, accidents, traffic control, PoI, etc. Existing approaches either concatenate the representation vectors or apply element-wise sum or product. These solutions may produce a joint representation but cannot capture the complex relationships between ST data and external factors as they do not have any explicit objective function to find correlations between them. Thus, *fusing external factors with the ST data to improve the overall model's performance is still remains a challenging task*.

To handle the above-mentioned challenges, we propose a deep spatial-temporal generative model (named D-GAN) for more accurate ST prediction in multiple time steps. D-GAN provides a general design to handle spatio-temporal prediction for different urban applications. D-GAN adopts a GANs-based structure and jointly learns generation and variational inference of data to produce stochastic predictions of the future accurately. D-GAN learns stochastic latent variables by optimizing variational lower bound combined with the adversarial loss. This allows to learn stochastic posterior distributions of data for handling the multimodality and the joint distribution of ST data. This is inspired by VAE-GANs which has been applied for image generation [21], multimodal image-to-image translation [22], and video prediction [23]. However, applying image generation and video prediction approaches directly are not applicable for predicting ST data of urban applications, such as taxi demand, crowd flow, etc. An image generation process can consider the large changes in appearance between the input and output image, but it cannot adequately handle spatial variations. Video prediction process can take spatial changes into account, but appearance remains largely the same from image-to-image. In addition, the prediction of next image in image sequences is highly dependent on its previous image. We extend VAE-GANs to stochastic ST data prediction for urban applications, such as taxi demand, crowd flow, etc.

Our deep stochastic spatial-temporal generative model has two components. **First**, we propose a spatio-temporal correlation network to model spatio-temporal joint distribution of data and support a stochastic sampling of the latent variables for multiple plausible futures; This network also supports the fusion of external factors with ST data through explicit objective to improve the model performance by learning a shared representation and then reconstruct them from the learned shared representation to discover correlations among them. **Second**, we design a stochastic adversarial network to jointly learn generation and variational inference of ST data through implicit distribution modelling. To the best of our knowledge, we are the first to propose latent

variable model to successfully show the stochastic multi-frame spatio-temporal prediction on real-world data for urban applications, such as taxi demand, crowd flow, etc. We also propose to use the 3D ST maps with different temporal resolution as model input which allows to capture the complex temporal dynamics of ST data. In addition, we validate our proposed model, D-GAN on large-scale real-world datasets, including taxi data of New York City (NYC) and bike-sharing data of NYC. The comparisons with baseline methods show the effectiveness of our proposed model, D-GAN.

In summary, the main contributions of this paper can be concluded as follows:

- We for the first time propose a stochastic spatial-temporal generative model (named, D-GAN) to predict spatio-temporal data accurately in multiple time steps for urban applications, such as taxi demand, crowd flow, etc. D-GAN is highly flexible and extendable as it can be easily extended for a new ST problem with multiple data sources.
- In D-GAN model, the variational inference is combined with GANs to jointly capture ST correlations, underlying factors of variations and multimodality in the data. We also propose to use the 3D ST maps with different temporal resolution as model input for capturing the complex temporal dynamics of ST data.
- Results show that stochastically sampling the ST feature space to predict future ST dynamics are plausible. We demonstrate application of the learned model to challenging task, like taxi demand prediction on two large-scale real-world datasets. We extend proposed model for another challenging task, crowd flow prediction to show that our proposed solution provides a general design to handle spatio-temporal prediction for different urban applications. The results show that D-GAN is achieving more accurate performance than baseline methods for both applications.

The rest of this paper is organized as follows: Section 2 reviews the previous works on spatio-temporal prediction for urban applications and introduces the background of our proposed model. In Section 3, we present the formal definition of the studied problem. Section 4 introduces our proposed model and its different components. In Section 5, we describe the datasets, baselines and provide the implementation details of our proposed model. Evaluation and analysis are also shown in Section 5. Finally, Section 6 concludes the paper.

2 RELATED WORKS

Spatiotemporal Prediction in Urban Computing

Spatiotemporal predictive learning is a fundamental problem for data-driven urban management. In recent times, many effective statistics and deep learning based models have been proposed for spatio-temporal prediction. Autoregressive integrated moving average (ARIMA) and its variants have been widely applied for spatio-temporal prediction [24][25]. But these models are not able to capture spatial and temporal relations. [25] proposed a framework where predicted demand is a weighted ensemble of three prediction models. While, some researchers aim to predict travel speed and traffic volume on the road [26][27]. These methods predict traffic volume only for single or multiple road segments instead of a city. Then time-series based prediction approaches [28][29] are proposed to capture the spatial relations with the external context data, such as weather, holiday, etc. But still these models could not model the complex non-linear ST relationships.

Besides traditional time-series models, deep learning based models achieved a great success for the modelling of ST data. Some researchers used context data from multiple sources and modelled that data using a stack of several fully connected layers for traffic demand prediction [30], and taxi supply-demand gap [31]. These models do not consider spatial and temporal relations explicitly. Some researchers explored the CNN to capture spatial correlation for ST prediction [2][10]. While, some researchers used RNN to model temporal dependencies in the data [32]. Very recent studies [14] used RNN and attention for learning spatial correlations and temporal dependencies. [33] introduced ST factors in the gates of RNN. ST-ResNet [2] modeled temporal closeness, trend, and period using the residual neural network for predicting the crowd flow in a city. [3] proposed ST network for predicting demand, and considered ST temporal correlation and semantic variations. [34] introduced a cascade multiplicative unit to learn the dependencies between multiple frames for traffic flow prediction. [35] used the meta-learning to model diverse traffic flow from other auxiliary geographical information. Recently, [5] proposed a GANs based structure to model the crowd flow prediction.

In recent times, deep learning has made a remarkable progress in generating future outputs either by designing different network architectures or by proposing different learning techniques, such as adversarial loss. However, these deep neural networks based approaches have certain limitations: **(1)** Assume that correlations between spatial and temporal dependencies are weak and model them independently; **(2)** Existing deep learning based ST prediction approaches use deterministic models, such as feed-forward and RNN which assume that outcome is deterministic, i.e., cannot model multimodal data [20]; **(3)** Existing approaches either directly concatenate representation vectors of external factors with ST data or apply element-wise sum or product. This can generate a shared representation, but it cannot capture correlations among ST data and external factors; **(4)** Using adversarial learning with input noise can indeed generate better results but fails to adequately cover the space of possible futures, i.e., either fail to capture the full distribution of outcomes, or yield blurry generations, or both; **(5)** Existing approaches do not support an inference network to support the reasoning about data at an abstract level. This is important for policy makers to identify what contributes to the model's improvement.

The proposed D-GAN model addresses these limitations by combining variational inference with GANs to jointly model the spatio-temporal correlations and uncertainties of multi-frame prediction. D-GAN also supports the fusion of external factors through multimodal integration to improve the model learning.

Variational Autoencoder (VAE)

As the deterministic LSTM model fails to capture the multimodal nature of a data, we use the VAE [36] to learn the complex data distribution in which model estimates a probability for the possible future sequence *y* instead of a single outcome. To model the multimodality, a latent variable *z* (sampled from prior distribution) is used to capture the inherent uncertainty. An autoencoder is a member of neural network models which learns compressed latent variables of a high-dimensional data. VAE is one of the autoencoders based on the Variational Bayesian and graphical model concept. VAE maps the input into a distribution rather than into a fixed vector. In the multivariate Gaussian case, model is trained by learning the mean (μ) and variance (σ) of the data distribution explicitly using reparameterization trick, while the stochasticity remains in the random variable $\epsilon \sim \mathcal{N}(0, I)$.

VAE uses deep learning with statistical inference for representing a data point in a latent space [36] and experiences the complexity in the approximation of intractable probabilistic computations. In addition, these generative models are trained by maximizing training data likelihood where likelihood-based methods go through the curse of dimensionality in many datasets, such as image, video. To handle the abovementioned

issues, [20] proposed Generative Adversarial Nets (GANs), an alternative training methodology to generative models. GANs is a novel class of deep generative models in which backpropagation is used for training to evade the issues associated with MCMC training [37].

Generative Adversarial Networks (GANs)

Recently, GANs has gained a lot of attention for generating realistic images as it avoids the difficulty related to maximum likelihood learning. GANs works with multimodal outputs and learns rich distributions implicitly over images and data which are hard to model with an explicit likelihood. GANs uses the concept of a non-cooperative game in which two networks, a generator (G) and a discriminator (D), are trained to play against each other as shown in Figure 3. G takes latent vector z from a prior distribution p_z as input and outputs a sample G(z) with the goal of bringing G(z) as close as possible to D(x) where data x is drawn from the true data distribution, p_{data}. At the same time, D tries to avoid getting fooled by G. A GANs model is well trained when equilibrium is achieved between D and G, and D cannot distinguish whether a sample is generated by G or generated from the real data distribution.

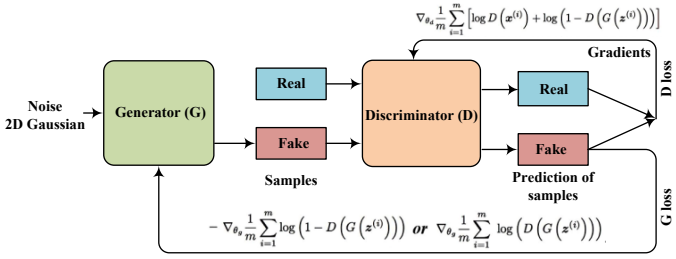

Figure 3. Basic GANs architecture

Basic GANs use two objective functions: (1) D minimizes the negative log-likelihood for binary classification; (2) G maximizes the probability of generated samples for being real. D parameters are denoted by θ_D, which are trained to maximize the loss to distinguish between the real and fake samples. G parameters are denoted by θ_G which are optimized such that the D is not able to distinguish between real and fake samples generated by G. θ_G is trained to minimize the same loss that θ_D is maximizing. Hence, it is a zero-sum game where players compete with each other. The following minimax objective applied for training G and D models jointly via solving:

$$\min_{\theta_G} \max_{\theta_D} V(G,D) = \min_{G} \max_{D} \mathbb{E}_{x \sim p_{data}}[\log D(x)] + \mathbb{E}_{z \sim p_z}\left[\log\left(1 - D\big(G(\mathrm{z})\big)\right)\right] \quad (1)$$

$V(G,D)$ is a binary cross entropy function, commonly used in binary classification problems. In Eq. 1, for updating the model parameters, training of G and D are performed by backpropagating the loss via their respective models. With the enough capability at D and G, and sufficient training iterations, G will be able to transform a simple prior distribution p_g to more complex distributions, i.e., p_g converges to p_{data}, such as $p_g = p_{data}$. In practice, the players are represented with deep neural nets and updates are made in parameter space.

Autoencoders learn the relationship between data and its latent code directly (i.e., explicit), while GANs learn to generate samples indirectly (i.e., implicit) [37]. D-GAN composes of both VAE and GANs to handle the ST data prediction problem in an effective way.

3 PRELIMINARY

In this section, we define the notations and the studied problem.

3.1 Notations

Region. There are many ways to define a location w.r.to different granularities and semantic meanings. In this study, we divide a city based on longitude and latitude into 2D non-overlapping grid map of $m \times n$ size, where a grid represents a region r in a city and a grid map represents a ST map (S) where $S = \{r_1, r_2, \dots, r_{mn}\}$.

Measurements. There are different types of measurements for a region that can be used for various ST applications, such as taxi demand, crowd flows prediction, and air quality prediction. In this study, we first take a case study of on-demand service prediction in a city. The taxi demand at a region r_i ($i \in [1, mn]$) in a given period t is defined as the number of taxis starting their trips during this time period. To show that our proposed solution provides a general design to handle spatio-temporal prediction for different urban applications, we take case study of another ST application, crowd flow prediction. The crowd flow can be categorized as inflow and outflow. The inflow and outflow of crowd at a region r_i ($i \in [1, mn]$) in a given period t is defined as the number of crowds coming in and going out, respectively. Two ST maps are maintained at a time interval t, one for inflow and another for outflow. To have the general problem statement, we use the demand values and crowd in-flow/outflow values V as measurements.

External Factors (ExF). These represent the following information: weather data (e.g., rainy, sunny, etc.), PoI information and time meta (e.g., time of day, day of week, holidays).

Problem statement. Given a sequence of T observations $V^S_{Given} = [V^S_1, V^S_2, \dots, V^S_T]$ and a sequence of external factors $Ex\mathcal{F}^S = [Ex\mathcal{F}^S_1, Ex\mathcal{F}^S_2, \dots, Ex\mathcal{F}^S_T]$ where V^S_i and $Ex\mathcal{F}^S_i \in R^D$ is a D dimensional vector representing the observation at time t, we aim to build a model that is capable of predicting future ST maps, $V^S_{Predicted} = [\mathcal{Y}^S_1, \mathcal{Y}^S_2, \dots, \mathcal{Y}^S_T]$, where $\mathcal{Y}^S_i \in R^D$ represents the predicted t^{th} step in the future.

4 THE PROPOSED D-GAN MODEL

4.1 D-GAN Overview

We propose a novel deep generative model which jointly learns dynamic ST correlation, stochasticity in the data, fusion of multi-source data for multi-frame prediction. The overall architecture of the proposed D-GAN is illustrated in Figure 4. Our model comprises of two components:

- **Spatio-temporal Correlation Network.** It encodes the information from ST data samples and external factors sampled at different time interval into a ST latent vector and ExF latent vector, respectively. It learns a unified representation of the ST data and external factors which is used further for stochastic sampling of the latent variables for multiple plausible futures.
- **Stochastic Adversarial Network.** It uses the learned shared representation (sampled latent vector) to jointly learn generation and variational inference of data.

4.2 Spatio-temporal Correlation Network

The inputs to the spatio-temporal correlation network are ST maps and external factors and outputs a deep fused representation of latent vector sampled from the shared representation of ST maps and external factors. The spatio-temporal correlation network can be broken down into three sub-components: data mapping, encoder and external factors fusion.

Data mapping. We map the historical spatio-temporal data into the following categories which is passed to the encoder as an input.

- ST_h^d: A 3d-tensor having a ST data of a day where each ST map within a tensor contains ST data of an hour.
- ST_h^w: A 3d-tensor having a ST data of a week where each ST map within a tensor contains ST data of an hour.
- ST_d^w: A 3d-tensor having a ST data of a week where each ST map within a tensor contains ST data of a day.

These tensors are the input of the encoder which is designed to capture the spatial and temporal correlation.

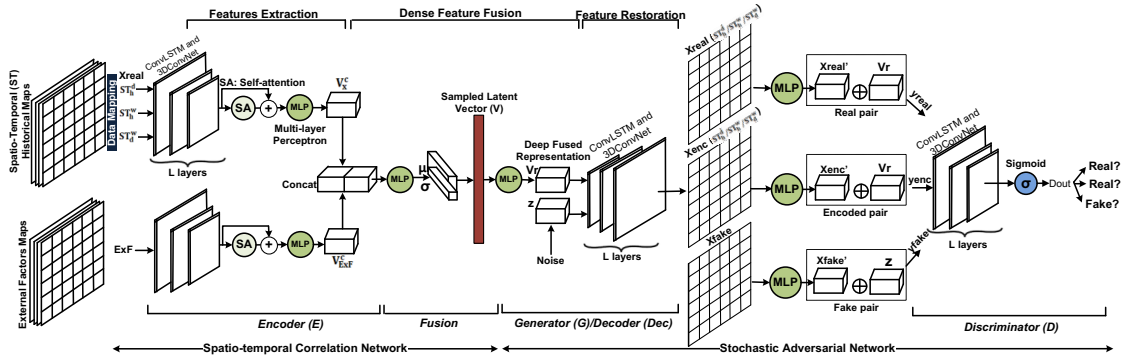

Figure 4. The network architecture of the proposed D-GAN. It contains two components: spatio-temporal correlation network and stochastic adversarial network

Encoder. We design an encoder (E) which encodes past ST data to learn ST correlation. The encoder can be broken down into a sub-component: fusion network. To extract features from the ST data, we use the ConvLSTM which is the convolution layer embedded within the LSTM. ConvLSTM uses convolution operation instead of matrix multiplication at each gate in the LSTM cell to capture the underlying spatial features in high-dimensional data. ConvLSTM includes both current input and the past states of its neighbours which allows it to model temporal dependencies. Thus, ConvLSTM can handle spatial dependencies by CNN and handle temporal dependencies by LSTM as follows:

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \\
f_t &= \sigma\big(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f\big) \\
C_t &= f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \\
o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \\
H_t &= o_t \circ \tanh(C_t)
\end{aligned}
\tag{2}
$$

where * is the convolutional operator, i_t is the input gate at time t, f_t is the forget gate, C_t is the cell state, o_t is the output gate and H_t is the hidden state at time t. Input gate control what to feed, forget gate decides what to forget and output gate chooses the output. Similarly, i_{t-1} is the input gate, f_{t-1} is the forget gate, C_{t-1} is the cell state, o_{t-1} is the output gate and H_{t-1} is the hidden state at time $t-1$. σ and tanh are the sigmoid and hyperbolic tangent activation functions, respectively. W is the weight and b is the bias.

We employ multiple stack of ConvLSTM units to extract spatial and temporal features. We pass the ConvLSTM output through a 3D-ConvNet, a self-attention layer [38] and a multi-layer perceptron (MLP) to produce a condensed feature vector $\mathbf{V}_x$, i.e.,

$$\mathbf{V}_x = \mathrm{MLP}\left(\mathrm{Attention}\left(\mathrm{ConvNet3D}\left(\mathrm{ConvLSTM}^{\mathrm{L}}(\mathbf{x}_{real})\right)\right)\right), \tag{3}$$

where $\mathbf{x}_{real}$ is a real data space either ST_h^d or ST_d^w or ST_h^w, $\mathbf{V}_x$ is the extracted feature vector of $\mathbf{x}_{real}$, and L is the number of ConvLSTM layers. ConvLSTM neural network is able to capture long-term trends in the ST map sequences while 3D-ConvNet captures local spatial dependencies. 3D-ConvNets can capture better short-term demand/crowd flow fluctuations and improves overall model's generalization abilities as it maintains the relationship between neighboring input points by sharing weights across different locations in the input and ST locality in feature representations. We use a self-attention layer to capture long-range spatial dependencies. Spatio-temporal data prediction in urban applications, such as taxi demand prediction, crowd flow prediction, a particular region can be connected to a distant region in a city. Existing approaches use the convolution to model the dependencies across different city regions. However, convolution operator has a local receptive field which requires several convolutional layers to compute the long-range dependencies. As a solution, size of the convolution kernels can be increased but it loses the computational and statistical efficiency obtained by using local convolutional structure. To handle this issue, self-attention [39]–[41] has been proposed to model the long-range dependencies with the computational and statistical efficiency. Unlike convolution, self-attention is used to model long-range, multi-level across image regions. The module of self-attention computes the response at a position as a weighted sum of the features at all positions where the weights or attention vectors are computed with only a small computational cost. Inspired from the mentioned benefits, we use the self-attention with the G [38] in which long-range dependencies are calculated for each region in the ST map.

We use the same layered architecture of encoder for different inputs (ST_h^d, ST_h^w and ST_d^w). The outputs of encoders of different inputs are concatenated $\mathbf{V}_x^c$ and passed through an MLP to have the latent vector of ST data. An encoder used to extract features of ST data is called as ST encoder.

Fusion. It is used to further boost the model performance by considering the influence of different external factors. External factors, such as PoI, weather, time, etc., have high effect on the ST applications, such as traffic prediction, demand prediction. We design a general fusion network to incorporate several external factors from different domains.

For each external factor, a feature map is extracted using a stack of ConvLSTM layers, a ConvNet3D, a self-attention layer and an MLP, similar as the ST Encoder. Then, the learned auxiliary feature vector is represented as $\mathbf{V}_{\mathbf{ExF}^i}$, $i = 1, 2, \ldots, w$, where w is total number of external factors to be used. The extracted condensed feature vectors are concatenated to obtain more precise feature representation for better performance as follows:

$$\mathbf{V}_{\mathbf{ExF}}^{C} = \left(\left(\left(\mathbf{V}_{\mathbf{ExF}^i}\right) \oplus \mathbf{V}_{\mathbf{ExF}^{i+1}}\right) \ldots \oplus \mathbf{V}_{\mathbf{ExF}^w}\right) \tag{4}$$

where $\oplus$ represents concatenation. An encoder used to extract features of external factors data is called as ExF encoder. The spatio-temporal feature representation $\mathbf{V}_x^c$ and the external factors feature representation $\mathbf{V}_{\mathrm{ExF}}^c$ are then concatenated and passed through an MLP to form the shared representation $\mathbf{V}_s$. Whole latent vector $\mathbf{V}_s$ cannot capture the distribution of ST correlation for each pixel location. To learn the variations in the data, we use a Variational Bayesian method with the multivariate Gaussian assumption and variational lower bound loss function where model calculates the mean (μ) and variance (σ) of the distribution explicitly from the shared representation $\mathbf{V}_s$ as follows:

$$\mathbf{V} = \mu + \sigma \odot \epsilon, \text{where } \epsilon \sim \mathcal{N}(0,\ I), \tag{5}$$

Where $\mathbf{V}$ is the final reparameterized multimodal representation, ϵ is an auxiliary independent random variable sampled from a previous distribution, and $\odot$ is the element-wise product. The deep fused representation $\mathbf{V}$ from the fusion network is passed to generator G.

We denote the spatio-temporal correlation network as $G_{STN}(X, \theta_{STN})$ where θ_{STN} denotes all the parameters to be learned in encoder, and $\boldsymbol{x}$ represents the set of ST data as input (i.e., ST and ExFs data). Hence, the output of the spatio-temporal correlation network for the ST and ExFs data $\boldsymbol{x}$ is the multimodal representation:

$$\mathbf{V} = G_{STN}(\boldsymbol{x}, \theta_{STN}) \tag{6}$$

4.3 Stochastic Adversarial Network

Our proposed D-GAN integrates the directed graphical model, i.e., VAE with GANs to learn the ST correlations and stochastic elements that exists within ST data. The stochastic adversarial learning network can be broken down into two sub-components: a generator (G) (i.e., decoder (Dec)) and a discriminator (D).

Generator (G)/Decoder (Dec). An MLP is applied on the multimodal representation $\mathbf{V}$ to get the deep fused representation $\mathbf{V_r}$ of ST and ExFs data. Then, G decodes the deep fused representation to reconstruct the input ST map in original size using an MLP and stacked ConvLSTM layers and a 3D-ConvNet.

$$\mathbf{x}_{enc} = \mathrm{ConvNet3D}\left(\mathrm{ConvLSTM}^{\mathrm{L}}(\mathbf{V_r})\right), \tag{7}$$

where x_{enc} is the reconstructed ST map of $\mathbf{x}_{real}$ and $\mathbf{x}_{enc}^{ExF}$ is the reconstructed map of $\mathbf{x}_{ExF}$. Furthermore, a noise vector ($\mathbf{z}$) is passed to G as input for generating reconstructed ST map of noise vector $\mathbf{x}_{fake}$. After training, G is used to generate samples similar to real data.

We denote the decoder as $G_{dec}(V, \theta_{dec})$ where θ_{dec} denotes all the parameters in the decoder. Hence, the output of the decoder for the ST and ExFs data, is reconstructed feature of the ST data, $\mathbf{x}_{real}$ and $\mathbf{x}_{ExF}$.

$$\mathbf{x}_{enc}, \mathbf{x}_{enc}^{ExF} = G_{dec}(V, \theta_{dec}) \tag{8}$$

Discriminator (D). D learns to determine whether a generated ST map is from the ground truth or produced by G. For the D's input, we concatenate the deep fused representation $\mathbf{V_r}$ with its generated ST map latent $\mathbf{x}_{enc'}$ and real ST map latent $\mathbf{x}_{real'}$ to jointly learn the latent code and data space in a pair as we notice that it supports fast convergence, better learning and high training stability, i.e., y_{enc} = $[\mathbf{x}_{enc'}, \mathbf{V_r}]$ is the encoded pair and y_{real} = $[\mathbf{x}_{real'}, \mathbf{V_r}]$ is the real pair. We also generate y_{fake} = $[\mathbf{x}_{fake'}, \mathbf{z}]$, a fake pair, where $\mathbf{z}$ is the noise feature vector. Then, we use similar stacked ConvLSTM layers and a ConvNet3D layer as follows:

$$D_{out} = \sigma\Big(\mathrm{ConvNet3D}\left(\mathrm{ConvLSTM}^{\mathrm{L}}(y)\right)\Big), \tag{9}$$

where y is y_{real}, y_{enc} and y_{fake}, and σ(·) is the sigmoid function to transfer convolutional output into probability. D_{out} is the predicted probability of input y being real or fake. G and D are trained simultaneously till D cannot discriminate the ST map generated by G with ST map generated from real data.

We denote the discriminator D as $G_{disc}(y, \theta_{disc})$, where θ_{disc} denotes all the parameters in the discriminator. The output of D for the reconstructed ST data and real data, is the probability of input y being real or fake.

$$D_{out} = G_{disc}(y, \theta_{disc}) \tag{10}$$

We can view the value of D_{out} as a label 1 means D detected input y as real data and 0 otherwise.

4.4 Learning Process

In this work, we propose a general model for different prediction tasks, such as regression and classification, on different types of spatio-temporal data. For different types of prediction tasks, it is required to identify different loss functions accordingly.

Variational loss comprises of two losses as follows: 1) KL divergence (D_{kl}). It measures the divergence between two probability distributions. 2) Reconstruction loss ($\mathcal{L}_{\mathrm{rec}}$). It calculates element-wise deviations between the ground truth and the reconstructed ST map to find the local differences between grids ($\mathcal{L}_{rec-x}$) and similar for external factors ($\mathcal{L}_{rec-ExF}$). $\mathcal{L}_{rec}$ is defined as the element-wise L2-norm. KL divergence minimization means here is to optimize the probability distribution parameters (μ and σ) to closely match that of the target distribution. These can be calculated as follows:

$$\mathcal{L}_{rec-x} = \frac{1}{mn} \|\mathbf{x}_{real} - \mathbf{x}_{enc}\|_2 \tag{11}$$

$$\mathcal{L}_{rec-ExF} = \frac{1}{mn} \left\|\mathbf{x}_{ExF} - \mathbf{x}_{enc}^{ExF}\right\|_2 \tag{12}$$

$$\mathcal{L}_{kl} = \frac{1}{2}\sum_{i=1}^{p}\left(\mu_i^2 + \sigma_i^2 - log(\sigma_i) - 1\right) \tag{13}$$

where mn is the total number of regions in a ST map, p is the dimensionality of multimodal features and D_{kl} is the KL-divergence. We minimize the VAE loss by seeking optimal parameters $\hat{\theta}_{STN}$ and $\hat{\theta}_{dec}$ and this can be represented as follows:

$$(\theta_{STN}^*, \theta_{dec}^*) = \underset{\theta_{STN}, \theta_{dec}}{\operatorname{argmin}} \left(\mathcal{L}_{rec-x} + \mathcal{L}_{rec-ExF} + \mathcal{L}_{kl}\right) \tag{14}$$

We use the least squares loss function instead of binary cross entropy used in GANs to evaluate the difference. The adversarial loss is used to find the equilibrium between G and D during the adversarial training process. In D-GAN, the adversarial loss of D ($\mathcal{L}_{\mathrm{GAN}}^{\mathrm{D}}$) is as follows:

$$\mathcal{L}_{GAN}^{D} = \|\mathrm{D}(y_{real}) - 1\|_2^2 + \left\|\mathrm{D}\left(y_{fake}\right) - 0\right\|_2^2 + \|\mathrm{D}(y_{enc}) - 1\|_2^2, \tag{15}$$

On the other hand, G's aim is to generate real-looking samples w.r.to D, so to minimize the G loss ($\mathcal{L}_{GAN}^{G}$), G tries to reduce the difference between $\mathrm{D}(y_{enc})$ and true label, and $\mathrm{D}(y_{fake})$ and true label as shown in Eq 15.

$$\mathcal{L}_{GAN}^{G} = \left\|\mathrm{D}\left(y_{fake}\right) - 1\right\|_2^2 + \|\mathrm{D}(y_{enc}) - 1\|_2^2, \tag{16}$$

The overall GANs loss can be calculated as follows:

$$\mathcal{L}_{GAN} = [(D(y_{real}) - 1)^2] + \left(D\left(y_{fake}\right) - 0\right)^2 + [(D(y_{enc}) - 1)^2] \tag{17}$$

We minimize this loss by seeking the optimal parameters $\hat{\theta}_{disc}$ and $\hat{\theta}_{dec}$ and can be represented as follows.

$$(\theta_{dec}^{*}, \theta_{disc}^{*}) = \arg \min_{\theta_{dec}} \max_{\theta_{disc}} \mathcal{L}_{GAN} \tag{18}$$

We jointly train the spatio-temporal correlation network (STN), decoder and discriminator. Thus, the final loss of D-GAN is as follows:

$$\mathcal{L}_{final}(\theta_{STN}, \theta_{dec}, \theta_{disc}) = \lambda_r \mathcal{L}_{rec-x} + \lambda_e \mathcal{L}_{rec-ExF} + \lambda_{kl} \mathcal{L}_{kl} + \lambda_g \mathcal{L}_{GAN} \tag{19}$$

where the hyperparameter λ control the relative importance of each term. The optimal parameters can then be calculated by minimizing the final loss as follows.

$$(\theta_{STN}^{*}, \theta_{dec}^{*}, \theta_{disc}^{*}) = \underset{\theta_{STN}, \theta_{dec}}{\operatorname{argmin}} \max_{\theta_{disc}} \mathcal{L}_{final}(\theta_{STN}, \theta_{dec}, \theta_{disc}) \tag{20}$$

5 EVALUATION

We demonstrate application of the learned model to challenging task, like taxi demand prediction on two large-scale real-world datasets. The main aim of demand prediction task is to learn an accurate model to predict the total number of requests for a particular service in each grid of ST map during each time slot where a time slot can be an hour, or a day, or a week. We extend proposed model for another challenging task, crowd flow prediction to show that our proposed solution provides a general design to handle spatio-temporal prediction for different urban applications. The main aim of crowd flow prediction task is to learn an accurate model to predict the total number of inflow/outflow in each grid of corresponding ST map during each time slot.

In this section, we shall demonstrate the effectiveness of the proposed D-GAN on large-scale real-world datasets: the yellow taxi dataset[1] and the bike trip dataset[2] in New York. We will first introduce the datasets, the baselines, and the experiment settings. Then, we will show the experiment results. In particular, we aim to answer the following research questions:

- Q1: How does our D-GAN model perform compared to the baseline approaches?
- Q2: What is the performance of D-GAN's variants with different combinations?
- Q3: What is the performance of D-GAN with the external factors' fusion?
- Q4: What is the impact of the proposed two-level temporal correlation scheme on D-GAN's performance?

5.1 Datasets

The details of the two datasets used for evaluation are described as follows:

- **TaxiNYC dataset.** This dataset contains the 2,062,262 taxi trip records of yellow taxis from Janunary 2016 to June 2016. Each trip record includes the coordinates and times of pickup and dropoff events.
- **CitiBikeNYC dataset.** The bike data are collected from NYC CitiBike system available for the bike sharing service from January 2016 to June 2016. This dataset contains the 4,500,000 taxi trip records where each record includes the coordinates and times of pick-up and drop-off events.

[1] http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.
[2] https://www.citibikenyc.com/system-data

- **BikeNYC dataset** [42]**.** Generated from the NYC bike trajectory data for 182 days, this dataset contains 4,392 traffic flow maps with a time interval of one hour. The given input and output in the dataset are (16, 8,10) and output is (16, 8, 2), respectively. As for external factors, 20 holiday categories are recorded. The first 172 days are used for training and the data of the last ten days are chosen to be the testing set.

Urban areas are divided into different regions where these regions may have different functions, such as shopping mall, airport, business function [43]. Different functional areas have different traffic flow and taxi demand patterns. For example, students commute from residential areas to their university in the morning and return home in the evening. [44] mentioned that in some prepared datasets NYC area is partitioned into regions without considering functional areas; therefore, some regions are water area where demand is always zero which causes to decrease the mean error and affect the evaluation of algorithm performance. Considering this issue, we study different functional areas in NYC dataset and partition NYC area into 100 regions using K-means clustering method [45] on average historical demand observations. After that, we drop the region having very low demand most of the time and get 81 active regions. In other words, the NYC area clustered into 9×9 non-overlapping regions using k-means which represents as an ST map.

To show that our proposed solution can work on a more fine-grained manner, we extend our analysis for the (16, 8) map of widely used preprocessed BikeNYC dataset [36] for crowd flow prediction. In the given dataset, the dimension of input and output ST map is different, (16, 8, 10) and (16, 8, 2), respectively, therefore, we extend our proposed solution to the conditional D-GAN in which output of G is conditioned with the input.

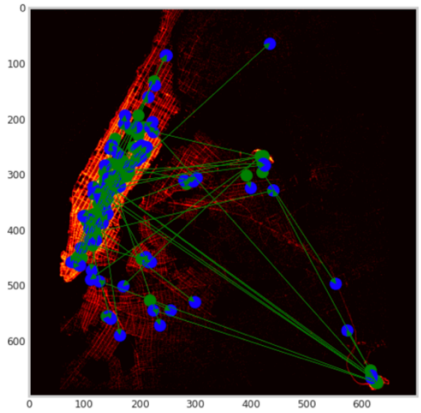

Figure 6(a). Clusters distribution of TaxiNYC dataset. Green color circle shows cluster of pickup location while Blue color shows cluster of drop-off locations. Green lines show the trips

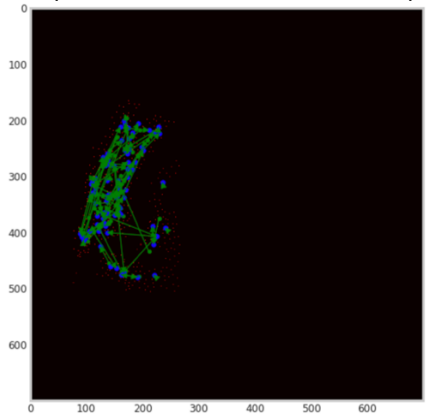

Figure 6(b). Clusters distribution of CitiBikeNYC dataset. Green color circle shows cluster of pickup location while Blue color shows cluster of drop-off locations. Red lines show the trips

In addition, we use the external factors, such as Point of Interest (PoI), weather data and weekend/weekday with the historical data. We also use the publicly available PoI data of New York[3]. There are 18,912 POIs in total, and it includes the POIs of the following facility domains: residential, education facility, culture facility, recreational facility, social services, transportation facility, commercial, government facility, religious institution, health services, public safety, water, and miscellaneous. We also use weather information, including the weather conditions of 16 types (rainy, snowy, sunny, etc.), temperature, and so on. Figure 6(a) and (b) show

[3] https://data.cityofnewyork.us/City-Government/Points-Of-Interest/rxuy-2muj/data

the clusters distribution of the TaxiNYC and CitiBikeNYC datasets, respectively. We can notice that trips' distribution in the taxi dataset is diverse in compared to the Bike dataset.

5.2 Evaluation Metrics

We use the widely adapted Rooted Mean Square Error (RMSE) and Mean Absolute Error (MAE) as the evaluation metrics as follows:

$$\text{RMSE} = \sqrt{\frac{1}{z}\sum_i^z(\hat{y}_i - y_i)^2}, \text{MAE} = \frac{1}{z}\sum_i^z|\hat{y}_i - y_i|$$

where $\hat{y}_i$ is prediction ST maps, y is real ST map, and z indicates number of samples used for validation.

5.3 Baselines

To illustrate the effectiveness of our D-GAN model, we compare it with the following baselines, both traditional statistical models and the state-of-the-art deep learning models and tune parameters for all methods.

- **Moving Average (MA):** It predicts the data value using average values of previous data values at the location given in the same relative time interval.
- **Autoregressive integrated Moving Average (ARIMA):** A well-known model which combines MA and autoregressive components for time-series data modelling.
- **Linear Regression (LR):** We use ordinary least squares regression (OLSR) model to estimate the relationship between multiple variables.
- **XGBOOST (XGB):** It is an ensemble learning in which many ML models are trained at once for better performance. The number of trees set to 80, and the maximum depth is 4.
- **Long Short Term Memory Neural Network (LSTM):** A neural network of a LSTM with a fully connected layer. The hidden unit is set as 64 and learning rate set to 0.001.
- **Convolutional Neural Network (CNN):** We use the two convolutional layers followed by a MaxPooling layer and a fully connected layer.
- **ST-ResNet** [2]**:** A deep convolutional based residual networks used for the grid-based traffic flow prediction. External factors are fused with the learned ST features after applying two fully connected layers.
- **DMVST-Net** [3]**:** A deep multi-view ST neural network, i.e., temporal, spatial and semantic view for grid-based prediction. External factors are directly concatenated with the learned ST features by CNN.
- **STGCN** [46]**:** A spatial-temporal Graph Convolutional Network which combines graph convolution with gated temporal convolution.

5.4 Parameters Setting

- **Preprocessing.** We use the Min-Max normalization [0, 1] on the training set to normalize the demand values. After training, we apply an inverse of the Min-Max transformation to recover the actual demand values. We choose first 90% of the data as training data and the remaining 10% is used for the testing in the case of demand prediction.

- **Network architecture.** We use four ConvLSTM layers with 32/16/8/4 number of filters and 3×3 size of filters. We use the Batch normalization (BN) after each ConvLSTM layer. After ConvLSTM layers, we use the Conv3D with Max-pool (2x2x2) and BN. The output of Conv3D is passed to the two fully connected (FC) layers. E and D use the same layer architecture as discussed for downsampling, while G shares the same network for upsampling.
- **Hyperparameters.** The batch size is 32 and number of epochs are 500.
- **Activation function.** We apply LeakyReLU [47] for FC layers and Conv3D as the activation function and use Sigmoid function for D's output layer.
- **Optimization method.** We use the stochastic gradient descent (SGD) algorithm with learning rate 0.001, decay 1e-6, and momentum 0.9. To prevent overfitting, we apply the dropout method with probability 0.4 between two FC layers.
- **Experimental environment.** Our model is trained on two Tesla P100-PCIE GPU with 16 GB memory at 1.3285 GHz and an operating system of 64 bits in the University Research Facility in Big Data Analytics (UBDA) of the Hong Kong Polytechnic University. (UBDA website: https://www.polyu.edu.hk/ubda/). The programming environment is Keras with TensorFlow as backend.

Table 1: Performance comparison among different baseline methods

Models	Demand Prediction				Crowd Flow Prediction	
	TaxiNYC data		CitiBikeNYC data		BikeNYC data	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
HA	11.41	-	11.96	-	21.57	-
ARIMA	11.35	-	13.56	-	10.07	-
LR	6.65	4.74	10.32	6.43	8.17	4.33
XGB	5.85	4.29	9.62	6.21	7.78	4.12
LSTM	3.81	1.39	9.01	6.15	7.21	3.37
CNN	3.44	1.28	8.78	4.03	7.01	3.22
ST-Res-Net	3.38	1.74	8.14	4.78	6.37	2.95
DMVST-Net	3.30	1.49	7.06	3.47	6.11	2.86
STGCN	2.67	1.25	6.75	3.11	5.98	2.75
D-GAN (24 steps prediction)	**1.35**	**1.01**	**4.09**	**2.85**	**4.77**	**2.63**

5.5 Performance Comparison with Baselines (Q1)

We now compare D-GAN with the baseline methods in predicting the demand volume on NYC taxi and Bike datasets. Table 1 shows the performance of D-GAN in comparison to other baseline methods. From Table 1, we have the following main observations:

The proposed D-GAN performs best among other approaches in the case of demand prediction with lowest 1.35 and 4.09 RMSE, and 1.01 and 2.85 MAE for Taxi and Bike datasets, respectively. It achieves 49.2% and 39.4% (RMSE) relative improvement over the best performance among all baselines for Taxi and Bike datasets, respectively. In the case of crowd flow prediction, proposed solution also achieves lowest 4.77 RMSE and 2.63 MAE. It achieves 20.23% (RMSE) relative improvement over the best performance among all baselines. We believe that the benefits are credited to the effective design of the D-GAN – jointly capturing the dynamic spatial and temporal features and multimodality in the data. Classical approaches, such as HA and ARIMA perform quite poor with RMSE of 11.41 and 11.35, respectively, for taxi data, 11.96 and 13.56, respectively for bike

dataset in demand prediction, and 21.57 and 10.07, respectively in crowd flow prediction as these models predict demand using average values of previous demands given in same relative time interval. Regression methods, such as RF, and XGB achieve better performance than previous ones but these methods cannot capture the spatial and temporal variation trend at the same time. We extend our model comparison with the well-known deep learning methods, such as LSTM and CNN. LSTM cannot model spatial correlations while CNN cannot capture temporal dynamics. To model spatial correlations, ST-ResNet and DMVST-Net are the state-of-the-art baseline approaches for grid-based data prediction. Both approaches achieve limited improvements over the LSTM and CNN. On the other hand, STGCN also achieves limited improvements over the LSTM and CNN as it uses predicted conditions in the previous iterations as the historical observations for the next iterations; thus, errors accumulated in the prediction task. More layers in the network structure, the more error accumulates, particularly in the case of the long-term prediction. In addition, results obtained through traditional machine learning model use MSE which is a pixel-wise average over many slightly different possible solutions in the pixel space and cannot model multimodal data [20]. These approaches fuse the spatial and temporal features directly; therefore, they could not model the inherent relationships between spatial and temporal attributes. In addition, external factors are directly concatenated, therefore, correlation between external factors and ST data is missing.

In contrast, our proposed D-GAN model learns rich distributions implicitly over data which allows to learn the inherent relationships between spatial and temporal features and to model multimodal data. Results in Table 1 show that D-GAN achieves more accurate ST prediction in comparison to both traditional and deep learning based ST prediction models which verifies the advantage of learning inherent spatial and temporal relationships jointly, fusion of external factors through objective function and handling multimodality in the data. On the other hand, D-GAN prevents pixel-level error propagation by learning the data distribution implicitly. In our work, each future step is predicting 24 frames at a time and performance of 24 future frames is better than baselines approaches' single steps. We can say that existing approaches lack in modelling the temporal dynamics of ST content as input is 2D ST maps, while we propose to use the 3D ST maps with different temporal resolution as input to model the fine-grained temporal dynamics. 3D ST maps, i.e., video, which allows to capture the complex spatial dependencies evolving over time.

5.6 Evaluations on Variants of D-GAN (Q2)

We evaluate key components of our multimodal D-GAN to understand it well. We raised the following two questions: (i) Does the architecture of encoder and decoder play a crucial role in capturing dynamic spatial and temporal dependencies? (ii) Is the selection of loss plays a role for making accurate predictions? We consider five variants of the proposed method to answer these questions.

- **GANs-LSTM:** We use adversarial learning with LSTM layers to model the spatio-temporal data. The number of hidden units is 64.
- **GANs-CNN:** We use the adversarial learning with two convolutional and one fully connected layer.
- **GANs-CNN-LSTM:** We use the adversarial learning with two CNN layers, a fully connected layer and a LSTM layer.
- **GANs-ConvLSTM-n:** We use basic structure of GANs with four ConvLSTM and one 3DConvNet with same parameters. This architecture design is similar to D-GAN. The differences are as follows: (1) only ST_h^d data

is used as input which is a 3d-tensor having a ST data of a day where each ST map within a tensor contains ST data of an hour; (2) it could not handle the stochasticity in the data; (3) external factors are not fused.

Table 2: Comparison between different variants of D-GAN

D-GAN Variants	TaxiNYC data		CitiBikeNYC data	
	RMSE	MAE	RMSE	MAE
GANs-LSTM	2.72	1.63	5.14	4.46
GANs-CNN	3.27	2.45	5.33	4.16
GANs-CNN-LSTM	2.68	1.6	4.76	4.35
GANs-ConvLSTM-n	4.68	3.93	5.24	4.57
GANs-ConvLSTM-s	1.51	1.09	4.33	3.01
D-GAN-FS	8.89	7.65	8.54	7.03
D-GAN	**1.35**	**1.01**	**4.09**	**2.85**

- **GANs-ConvLSTM-s:** We use basic structure of GANs with four ConvLSTM and one 3DConvNet with same parameters. This architecture design is similar to D-GAN as it can handle the stochasticity in the data. The differences are as follows: (1) only ST_h^d data is used as input which is a 3d-tensor having a ST data of a day where each ST map within a tensor contains ST data of an hour; (2) external factors are not fused.
- **D-GAN-FS:** We use the feature similarity loss [21] instead of L2 loss.

Table 2 shows the performance comparison of D-GAN with other variants. We notice that full version of our developed model D-GAN achieves the best performance in most evaluation metrics across various settings. In particular, we summarize three key observations:

(i) Results in Table 2 show that GANs-based models are performing better than existing deep learning models as they are learning the sample generation from the probability distribution. Even though, GANs-LSTM and GANs-CNN models (Tensor size (TS): (9, 9)) are performing better than LSTM and CNN. GANs-CNN-LSTM is performing better than GANs-LSTM and GANs-CNN as it is able to capture both spatial and temporal features. But the D-GAN outperforms these models as basic GANs with input noise cannot adequately cover the space of possible futures, i.e., cannot generate high quality samples.

(ii) We then implement GANs-ConvLSTM-n (TS: (24, 9, 9, 1)) where each input feature of a ConvLSTM network is a three-dimensional ST tensor. Studies show that ConvLSTM network captures better ST correlation and consistently outperforms LSTM [13]. GANs-ConvLSTM-n's results show that accuracy is lower than GANs-LSTM and GANs-CNN. The reason can be that basic GANs cannot capture dynamic ST correlation for the high-dimensional data in comparison to low-dimensional data. We further extend our analysis and combine the variational inference with GANs (using ConvLSTM), named GANs-ConvLSTM-s which performs better than other variants as it can handle both highly-structured yet stochastic nature of ST data. Furthermore, we observe that D-GAN outperforms GANs-ConvLSTM-s which uses the same architecture with similar number of parameters, shows the effectiveness of external factors fusion and two-level temporal correlation for spatio-temporal prediction.

(iii) In the case of spatio-temporal data prediction in urban applications, such as taxi demand prediction, crowd flow prediction, a grid in the ST map (i.e., a pixel in the image), represents the number of Taxis/Bikes requests at a region in a time interval *t*. Our main focus is to minimize the pixel loss between generated and real data. To show the importance of the selected pixel loss, we use the feature loss in our proposed

solution, D-GAN. The result shows performance decreases significantly as two ST maps can have small feature loss even if they are significantly different in pixel-by-pixel comparison. We also note that D-GAN-FS does not convergence most of the time.

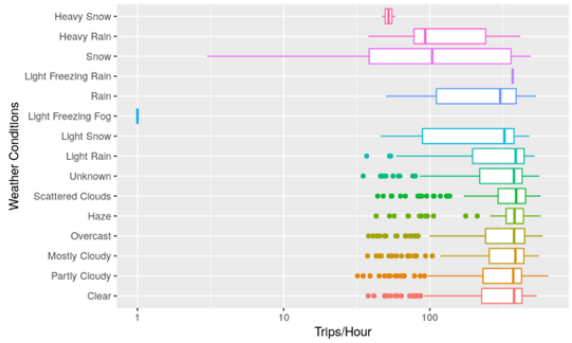

Figure 7. Impact of weather on the taxi trips

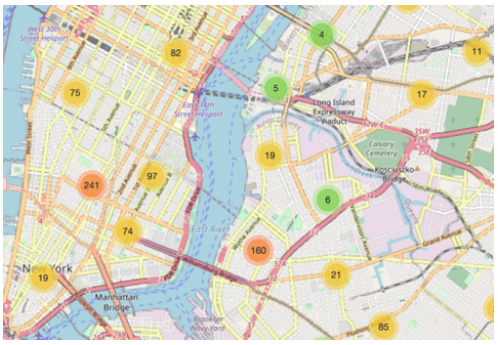

Figure 8. Distribution of PoIs in NYC area

Table 3 Impact of external factors on D-GAN

D-GAN Variants	TaxiNYC data		CitiBikeNYC data	
	RMSE	MAE	RMSE	MAE
D-GAN-ExF-n	1.41	1.09	4.21	2.93
D-GAN-ExF-c	1.40	1.08	4.17	2.90
D-GAN-ExF-1	1.41	1.09	4.21	2.93
D-GAN-ExF-2	1.41	1.09	4.21	2.92
D-GAN-ExF-3/D-GAN	**1.35**	**1.01**	**4.09**	**2.85**

5.7 Evaluation on External Factors Fusion (Q3)

We further evaluate the performance of D-GAN by fusing the additional data with the ST correlation module to enhance the overall predictive performance. We use PoI data (PoI), weekday/weekend (Dw) and weather data (W) as external factors (ExF). Figure 7 shows the impact of weather data on the taxi trips. Figure 8 shows the distribution of PoIs in the NYC area. We fuse the external factors one-by-one with the aim of answering the following two questions: (i) Is the fusion of external factors effective? (ii) What is the effect of a particular external factor on the D-GAN's performance? We consider five variants of the proposed method to answer these questions.

- **D-GAN-ExF-n:** External factors are not fused into the proposed model.
- **D-GAN-ExF-c:** External factors are directly concatenated with the proposed model.
- **D-GAN-ExF-1:** Only a PoI data is fused as external factor.
- **D-GAN-ExF-2:** PoI and Dw are fused into our proposed model.
- **D-GAN-ExF-3/D-GAN:** All external factors are fused into our proposed model. This is the complete version of our proposed model, D-GAN.

Table 3 shows the impact of external factors on the D-GAN performance. We notice that the full version of our developed D-GAN model achieves the best performance in most evaluation metrics. In particular, we summarize two key observations:

(i) In complete version of our prediction model, D-GAN, all external factors are fused, and results show that D-GAN outperforms all D-GAN variants in which either external factors are directly concatenated or not all external factors are fused. This shows that the external factors are useful and incorporating them does improve the model performance. Results also show that the improvement on the prediction model performance in the case of the D-GAN-ExF-c and D-GAN-ExF-n are approximately same. The reason is in the case of D-GAN-ExF-c that model does not have any explicit objective to discover correlations among different data sources.

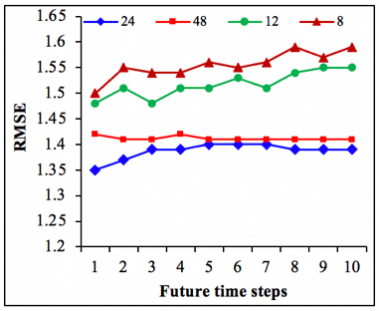

Figure 9. Impact of sequence length on RMSE and future time steps for TaxiNYC data. Each future step is predicting 24 frames at a time

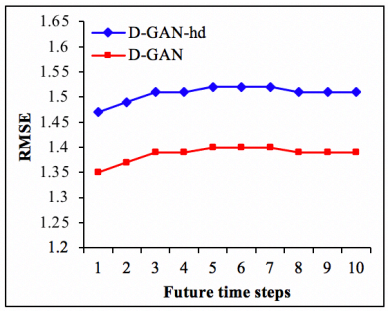

Figure 10. Impact of two-level temporal correlation on RMSE and future time steps for TaxiNYC data. Each future step is predicting 24 frames at a time

(ii) Result shows that D-GAN performs better than D-GAN-ExF-1 and D-GAN-ExF-2. The reason is that D-GAN jointly learns the ST correlations and underlying factors of variations by modelling ST data distribution; therefore, fusion of external factors, such as PoI (i.e., location) and Dw (i.e., time) are not effective. The improved predictive performance shows the advantage of our fusion network.

Table 4 Impact of Two-level Temporal Correlation on D-GAN

D-GAN Variants	TaxiNYC data		CitiBikeNYC data	
	RMSE	MAE	RMSE	MAE
D-GAN-hd	1.47	1.04	4.22	2.91
D-GAN	**1.35**	**1.01**	**4.09**	**2.85**

5.8 Effect of Two-level Temporal Correlation (Q4)

We propose the two-level temporal correlation approach for spatio-temporal data prediction to capture more comprehensive temporal dependencies. To investigate the effect of two-level temporal correlation, we study the performance of our D-GAN model with different data mapping.

- **D-GAN-hd:** The input to the proposed model is a 3d-tensor having a ST data of a day where each ST map within a tensor contains ST data of an hour.

Table 4 shows the effect of data mapping on the D-GAN performance. We notice that complete version of our proposed model D-GAN achieves the best performance in most evaluation metrics. In particular, we summarize two key observations:

(i) Here each future step is predicting 24 frames. First, we study how the sequence length of ST maps affects the model's performance (see Figure 3) (CitiBikeNYC dataset has approximately similar results for this

case, so we omit them). D-GAN-hd achieves the best performance for sequence length of 24 hours. While, for 8 and 12 hours, RMSE degraded. The reason can be that model could not capture temporal dependencies. As the sequence length increases to 48, at the step 1, the prediction error degrades slightly while in later steps, performance is approximately equal to the performance of 24 hours. The reason is that calculating longer temporal dependency means to train higher number of parameters, i.e., training becomes hard. In our work, each future step is predicting 24 frames at a time. This shows stochastically sampling the ST feature space to predict future ST dynamics are plausible.

(ii) We perform another experiment in which we passed three inputs to the proposed model. The input to the proposed model is a 3D-tensor having a ST data of hours/days/weeks where each ST map within a tensor contains ST data of an hour/day/week, i.e., full version of D-GAN. The results show that D-GAN performs better than D-GAN-hd (see Figure 10) (CitiBikeNYC dataset has approximately similar results for this case, so we omit them). The possible reason is that through the two-level temporal correlation, model is able to capture more fine-grained temporal dependencies.

Result shows that our full version of proposed model, D-GAN outperforms the D-GAN-hd which verifies that D-GAN can capture the long and complicated temporal dependencies.

6 CONCLUSION

In this paper, we propose a novel deep stochastic generative adversarial based network (named, D-GAN) for ST prediction in multiple time steps. D-GAN deeply captures the underlying ST data distribution implicitly for modelling ST correlations, underlying factors of variations and multimodality in the data. D-GAN comprises of two components: (1) a spatio-temporal correlation network to model ST correlations underlying factors of variations and multimodality in the data. It also includes a fusion network to learn correlations between the external factors and ST data using an explicit objective function; (2) a stochastic adversarial network to jointly learn generation and variational inference of ST data through implicit distribution modelling. We also propose to use two-level temporal correlation to capture fine-grained temporal dynamics. D-GAN is highly flexible and extendable as it can be easily extended for a new ST problem with multiple data sources. We evaluated the performance of our proposed model on a case study of demand prediction using two real-world datasets. To show that our proposed solution provides a general design to handle spatio-temporal prediction for different urban applications, we extend proposed model for another challenging task, crowd flow prediction. The results show that D-GAN is achieving more accurate performance than baseline methods for both applications. This research provides new insights for modelling the complex inherent spatial and temporal relationships and capturing variations in the data simultaneously.

Our model introduced a simple, but effective solution to multimodal spatio-temporal prediction of large continuous spaces and we expect that it will be useful in domains with uncertainty. In the future, we will study how to enrich this research by incorporating several ST data, such as bus/bike/truck data, mobile usage data, mobility data, etc., into account for collective prediction.

ACKNOWLEDGMENT

This work is supported by RGC Theme-based Research Scheme (Grant no: T41-603/20-R), RGC Collaborative Research Fund (Grant no: C5026-18G), and PolyU Internal Start-up Fund (Grant no: P0038876). The authors

would like to thank the PolyU University Research Facility in Big Data Analytics (UBDA) for providing servers for experiments in this paper.

REFERENCES

[1] Z. Chao, F. Pu, Y. Yin, B. Han, and X. Chen, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," *J. Sensors*, vol. 2018, pp. 1–9, 2018.

[2] J. Zhang, Y. Zheng, and D. Qi, "Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction," in *AAAI International Conference on Artificial Intelligence*, 2017.

[3] H. Yao *et al.*, "Deep multi-view spatial-temporal network for taxi demand prediction," in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2018.

[4] X. Wang *et al.*, "Spatio-Temporal Analysis and Prediction of Cellular Traffic in Metropolis," in *IEEE 25th International Conference on Network Protocols (ICNP)*, 2017.

[5] S. Wang, H. Chen, H. Peng, J. Cao, and Z. Huang, "SeqST-GAN: Seq2Seq Gener-ative Adversarial Nets for Multi-step Urban Crowd Flow Prediction," *ACM Trans. Spat. Algorithms Syst.*, vol. 6, no. 4, 2020.

[6] S. Wang, H. Miao, H. Chen, and Z. Huang, "Multi-task Adversarial Spatial-Temporal Networks for Crowd Flow Prediction," in *International Conference on Information and Knowledge Management, Proceedings*, 2020, pp. 1555–1564.

[7] H. Chen, S. Wang, Z. Deng, X. Zhang, and Z. Li, "FGST: Fine-grained spatial-temporal based regression for stationless bike traffic prediction," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11439 LNAI, pp. 265–279.

[8] B. Du *et al.*, "Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 972–985, Mar. 2020.

[9] Y. Zhang, S. Wang, B. Chen, J. Cao, and Z. Huang, "TrafficGAN: Network-Scale Deep Traffic Prediction with Generative Adversarial Nets," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 219–230, Jan. 2021.

[10] A. Nawaz, H. Zhiqiu, W. Senzhang, Y. Hussain, I. Khan, and Z. Khan, "Convolutional LSTM based transportation mode learning from raw GPS trajectories," *IET Intell. Transp. Syst.*, vol. 14, no. 6, pp. 570–577, Jun. 2020.

[11] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction," *Proc. AAAI Conf. Artif. Intell.*, 2019.

[12] H. Lin, W. Jia, Y. You, and Y. Sun, "Interpretable crowd flow prediction with spatial-temporal self-attention," *arXiv preprint arXiv:2002.09693*. 2020.

[13] X. Shi *et al.*, "Deep learning for precipitation nowcasting: A benchmark and a new model," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 5618–5628, 2017.

[14] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "Geoman: Multi-level attention networks for geo-sensory time series prediction," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2018-July, pp. 3428–3434, 2018.

[15] S. Wang, J. Cao, and P. S. Yu, "Deep learning for spatio-temporal data mining: a survey," *IEEE Trans. Knowl. Data Eng.*, Jun. 2020.

[16] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, 2016.

[17] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow Prediction in Spatio-Temporal Networks Based on Multitask Deep Learning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 3, pp. 468–478, 2020.

[18] X. Geng *et al.*, "Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 3656–3663, 2019.

[19] L. Theis, A. Van Den Oord, and M. Bethge, "A note on the evaluation of generative models," in *4th International Conference on Learning Representations, ICLR*, 2016.

[20] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, vol. 3, no. January, pp. 2672–2680.

[21] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *33rd Int. Conf. Mach. Learn. ICML*, vol. 4, pp. 2341–2349, Dec. 2016.

[22] J. Y. Zhu *et al.*, "Toward multimodal image-to-image translation," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. 1, pp. 466–477, 2017.

[23] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," *arXiv preprint arXiv:1804.01523*. 04-Apr-2018.

[24] X. Li *et al.*, “Prediction of urban human mobility using large-scale taxi traces and its applications,” *Front. Comput. Sci. China*, vol. 6, no. 1, pp. 111–121, 2012.
[25] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, “Predicting taxi-passenger demand using streaming data,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1393–1402, 2013.
[26] A. Abadi, T. Rajabioun, and P. A. Ioannou, “Traffic Flow Prediction for Road Transportation Networks With Limited Traffic Data,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 653–662, 2015.
[27] R. Silva, S. M. Kang, and E. M. Airoldi, “Predicting traffic volumes and estimating the effects of shocks in massive transportation systems,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 112, no. 18, pp. 5643–5648, 2015.
[28] F. Wu, H. Wang, and Z. Li, “Interpreting traffic dynamics using ubiquitous urban data,” in *SIGSPATIAL*, 2016.
[29] Y. Tong *et al.*, “The Simpler the Better: A unified approach to predicting original taxi demands based on large-scale online platforms,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1653–1662.
[30] H. Wei, Y. Wang, T. Wo, Y. Liu, and J. Xu, “ZEST: A hybrid model on predicting passenger demand for chauffeured car service,” in *Int. Conf. on Info. and Know. Mgmt, Proc.*, 2016, vol. 24-28-Oct, pp. 2203–08.
[31] D. Wang, W. Cao, J. Li, and J. Ye, “DeepSD: Supply-demand prediction for online car-hailing services using deep neural networks,” *Proc. - Int. Conf. Data Eng.*, pp. 243–254, 2017.
[32] Z. Cui, R. Ke, and Y. Wang, “Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction,” *arXiv preprint arXiv:1801.02143*. pp. 1–11, 2018.
[33] D. Kong and F. Wu, “HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction,” *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2018-July, pp. 2341–2347, 2018.
[34] Z. Xu, Y. Wang, M. Long, and J. Wang, “PredCNN: Predictive learning with cascade convolutions,” in *IJCAI International Joint Conference on Artificial Intelligence*, 2018, vol. 2018-July, pp. 2940–2947.
[35] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, “Urban traffic prediction from spatio-temporal data using deep meta learning,” in *ACM SIGKDD Int. Conf. on Know. Discov. & Data Min.*, 2019, pp. 1720–30.
[36] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.
[37] D. Saxena and J. Cao, “Generative Adversarial Networks (GANs): Challenges, Solutions, and Future Directions,” *arXiv Prepr. arXiv2005.00065*, 2020.
[38] A. Zhang, Han and Goodfellow, Ian and Metaxas, Dimitris and Odena, “Self-Attention Generative Adversarial Networks,” *arXiv Prepr. arXiv1805.08318*, 2018.
[39] J. Cheng, L. Dong, and M. Lapata, “Long Short-Term Memory-Networks for Machine Reading,” *EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc.*, pp. 551–561, Jan. 2016.
[40] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, “A Decomposable Attention Model for Natural Language Inference,” *EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc.*, pp. 2249–2255, Jun. 2016.
[41] A. Vaswani *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017, vol. 2017-Decem, pp. 5999–6009.
[42] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” in *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 2017, pp. 1655–1661.
[43] J. Yuan, Y. Zheng, and X. Xie, “Discovering regions of different functions in a city using human mobility and POIs,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 186–194.
[44] L. Liu *et al.*, “Dynamic Spatial-Temporal Representation Learning for Traffic Flow Prediction,” *IEEE Trans. Intell. Transp. Syst.*, pp. 1–15, 2020.
[45] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An Efficient k-Means Clustering Algorithm: Analysis and Implementation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, 2002.
[46] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” *arXiv preprint arXiv:1709.04875*. 2017.
[47] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” *ICML Work. Deep Learn. Audio, Speech Lang. Process.*, vol. 28, 2013.