Mitigating Cold-start Problem Using Cold Causal Demand Forecasting Model

Zahra Fatemi
Department of Computer Science
University of Illinois Chicago
zfatem2@uic.edu

Minh Huynh
Google
mahuynh@google.com

Elena Zheleva
Department of Computer Science
University of Illinois Chicago
ezheleva@uic.edu

Zamir Syed
Google
zamirs@google.com

Xiaojun Di
Google
xdi@google.com

Abstract

Forecasting multivariate time series data, which involves predicting future values of variables over time using historical data, has significant practical applications. Although deep learning-based models have shown promise in this field, they often fail to capture the causal relationship between dependent variables, leading to less accurate forecasts. Additionally, these models cannot handle the cold-start problem in time series data, where certain variables lack historical data, posing challenges in identifying dependencies among variables. To address these limitations, we introduce the *Cold Causal Demand Forecasting (CDF-cold)* framework that integrates causal inference with deep learning-based models to enhance the forecasting accuracy of multivariate time series data affected by the cold-start problem. To validate the effectiveness of the proposed approach, we collect 15 multivariate time-series datasets containing the network traffic of different Google data centers. Our experiments demonstrate that the CDF-cold framework outperforms state-of-the-art forecasting models in predicting future values of multivariate time series data suffering from cold-start problem.

1 Introduction

Time series forecasting plays a central role in a vast number of studies [44, 30, 32, 22]. The goal of time series forecasting is to use historical and current data to predict future values over a period of time. The applications of time series prediction are diverse and include climate and weather forecasting in geography [9, 14], traffic flow prediction in transportation [47, 8, 49], healthcare diagnosis in medical science [41, 3], and sales and stock prices prediction in economics [23, 31, 25, 4]. We specifically focus on forecasting the network traffic of different Google data centers as a motivating example. Each data center hosts multiple Google services with different network traffic and machine usage. Accurately forecasting the network traffic in these data centers is critical for efficient resource allocation and capacity planning, as well as ensuring a high-quality user experience.

In recent years, there has been extensive research and application of various time series forecasting methods. While classic forecasting methods like Univariate Autoregressive (AR), Univariate Moving Average (MA), Simple Exponential Smoothing (SES), and Autoregressive Integrated Moving Average (ARIMA) [2] have been widely studied, they are limited by their assumptions of linearity and aperiodicity of data. Moreover, these models fail to effectively forecast multivariate time series datasets, where each variable's behavior depends not only on its past values but also on the interactions

Temporal Graph Learning Workshop @ NeurIPS 2023, New Orleans.

with other variables. Recently, deep learning models have been developed to capture the complexity and nonlinearity in time series forecasting. Long Short-Term Memory (LSTM) is one of the prominent deep learning models used to extract dynamic information from time series data through the memory mechanism [36, 40].

While these approaches often perform well at capturing temporal patterns, they often overlook the interdependencies between different time series variables. The better the interdependencies among different time series are modeled, the more accurate the forecasting can be [47]. In the running example, the impact of different Google services on the network traffic can vary depending on the machine usage of the service. Recently, Graph Neural Networks (GNN) have been utilized to incorporate the topology structure and interdependencies among variables in forecasting tasks [42, 49]. In GNN models, each variable from a multivariate time series is represented as a node in a graph, and the edges between the nodes capture the interdependencies or relationships between the variables. By propagating information between neighboring nodes, Graph Neural Networks (GNNs) empower each variable in a multivariate time series to be aware of the influence of correlated variables. However, these models often lack the ability to effectively capture and understand causal relationships between variables. Incorporating causal knowledge into forecasting models enhances interpretability and provides insight into the factors that affect the target variable. Previous studies have attempted to quantitatively characterize the interdependencies between time series variables through causality [24]. Granger causality is one commonly used approach in time series analysis, particularly in economics [15]. Nevertheless, research has demonstrated that Granger causality cannot handle nonlinear relationships well, leading to spurious causality or the identification of false causal relationships [5].

While it is straightforward to train a forecasting model with past values of the time series, forecasting time series data with no historical data, known as cold-start forecasting, is challenging. In the absence of historical data, the forecasting model fails to capture and learn the inter-dependencies between new and existing variables, leading to inaccurate predictions of the target variables. For instance, when new Google services are introduced to a data center in the future, they may impact the total network traffic of the data center.

In this paper, we propose the Cold Causal Demand Forecasting (CDF-cold) framework that brings together causal inference and deep learning-based models to increase the accuracy of multivariate time series forecasting suffering from the cold-start problem. CDF-cold consists of two main components: 1) The Causal Demand Forecast component that exploits the causal relationship between different variables of a multivariate time series to learn a new representation for each variable based on the representation of other variables that causally impact it, and 2) The Cold-start forecasting component that leverages similarity-based approaches to alleviate the cold-start forecasting in time series data. While the idea of using deep learning models in time series forecasting is not novel, combining these models with causal inference to address the cold-start problem in multivariate time series forecasting is novel. To summarize, this paper makes the following contributions:

- We formulate the cold-start forecasting problem in multivariate time series datasets. In particular, we focus on datasets where future values for some variables correlated with the variables with no historical data are available in advance.
- We develop a similarity-based framework that incorporates causal relationships between variables in a deep learning model and addresses cold-start forecasting in time series data with no historical data.
- We evaluate the performance of our framework on 15 multivariate time series datasets from various Google data centers, containing network traffic, and machine usage of different Google services. Our results demonstrate that our proposed framework outperforms existing baselines in forecasting accuracy.

2 Related Work

Classical forecasting methods rely on statistical regression techniques to predict future values of variables based on historical information. The Autoregressive Integrated Moving Average (ARIMA) is one of the most widely used statistical methods for time series forecasting in non-seasonal time series. It combines Autoregression (AR), Moving Average (MA), and a differencing pre-processing step called integration (I) to make the time series stationary [18, 2]. In Vector Auto Regression

(VAR), each variable is a linear function of its past values and the past values of all the other variables. Despite their popularity, these models cannot capture nonlinear and complex temporal patterns among different time series, resulting in sub-optimal forecasts.

Recent studies have shown that deep learning methods consistently outperform classical methods in forecasting [26, 43, 17]. A line of research focuses on Recurrent Neural Network architectures for temporal forecasting applications [34, 29]. Salinas et al. [37] propose DeepAR, a method for obtaining accurate probabilistic forecasts using an autoregressive RNN model on a large number of time series datasets. Long Short-Term Memory (LSTM), which is a special type of RNN with additional features to memorize sequences of data, has gained lots of attention in traffic forecasting [51, 50]. A new line of research focuses on deploying Graph Neural Networks (GNNs) to integrate the dependency between variables in the forecasting model [45, 38, 7, 28]. Recently, attention mechanisms (e.g., transformers) have shown superior performance in time series forecasting due to their ability to handle long-term dependencies [10, 11, 8].

While there has been extensive research on developing deep-learning methods for time series forecasting, less attention has been paid to the impact of causal relationships between variables in a multivariate time series on forecasting accuracy. Granger causality is one the prominent approaches to identifying causal relationships in time series data [16, 13, 6, 12]. Nevertheless, prior studies have shown that Granger causality may lead to spurious or falsely detected causal relationships due to its inability to handle nonlinear relationships [5]. Xu et al. [47] develop an approach to identify causal relationships among variables and use this information in multivariate time series forecasting. A recent study introduced a method for causal discovery and forecasting in nonstationary time series data, with a primary focus on learning causal graphs from such data [20]. However, there's still a need to address the cold-start forecasting problem which is the focus of this paper.

3 Preliminaries

Suppose that we have N data centers, and each data center $L_i \in \mathbf{L}$ generates a multivariate time series recording A attributes (such as network traffic and machine usage of each Google service) over time. Let $\mathbf{X} \in \mathbb{R}^{N \times T \times A}$ denote the multivariate time series from all N data centers for a total of T time steps. We use $x_i \in \mathbb{R}^{T \times A}$ to represent the multivariate time series from data center L_i, $x_{i,T}$ to show the attributes from data center L_i at time T, and $x_{i,t}^j$ to denote the value of attribute a_j in data center L_i at time t.

The goal of time series forecasting is to learn a function F_θ that, at timestamp t, given the attributes of the past U timestamps from each data center L_i, predicts the values of the attributes in the future H timestamps. Formally, at timestamp t, the forecasting function F_θ predicts the values of all variables in data center L_i over the next H timestamps as:

$$(\hat{x}_{i,t+1}, \hat{x}_{i,t+2}, ..., \hat{x}_{i,t+H}) = F_\theta(x_{i,t-U+1}, x_{i,t-U+2}, ..., x_{i,t}) \tag{1}$$

where θ is a set of learnable model parameters of the forecasting model, H is the horizon ahead of the current timestamp, and $\hat{x}_{i,t-H+1}$ is the prediction for all variables in data center i at time $t - H + 1$.

3.1 Cold-start forecasting problem

Cold-start forecasting problem arises when there is no historical data available for time series data or when the available data is insufficient to make reliable predictions. The problem becomes even more complex when there are interdependencies between the time series impacted by the cold-start problem and other variables. One example of such interdependencies is when introducing a new Google service to a data center, which can have a substantial impact on the overall network traffic within the data center. This sudden change in the network traffic pattern can disrupt the existing relationships and correlations that machine learning models rely on for forecasting. As a result, accurately predicting future patterns becomes more challenging.

Conventional machine learning and neural network forecasting models face challenges when attempting to derive precise inferences from inadequate information. These models often assume that all variables have an equal impact on the target variable, which does not always hold true in real-world domains. For example, highly used services can exert a more substantial influence on the total network traffic of a data center compared to less utilized services. This discrepancy in impact can lead to inaccuracies in forecasting when conventional models are employed.

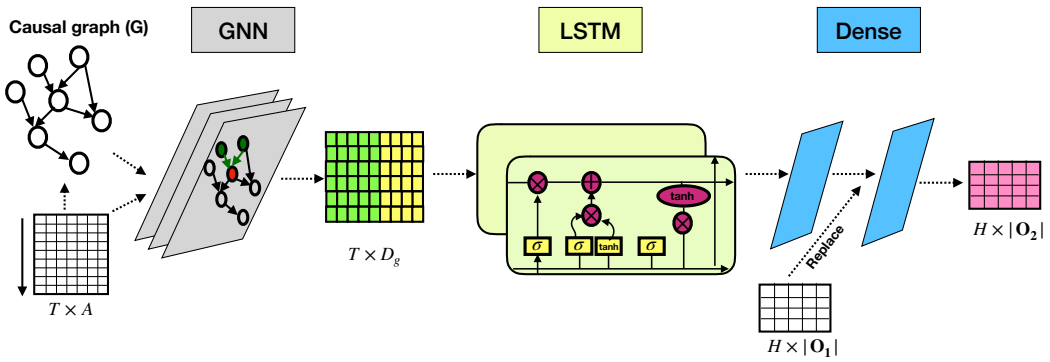


Figure 1: Illustration of Causal Demand Forecast (CDF) component. **Input**: a multivariate time series dataset and the causal graph representing the causal relationship between variables. **GNN**: generates a representation for each variable in each time point based on the variables causally impacting it. **LSTM**: generates a representation for each variable based on the historical data and the representation generated by the GNN layer. **Dense**: generate the forecasting for H horizons.

In some cases, future information for certain attributes that are correlated with time series impacted by the cold-start problem may be available in advance (e.g., machine usage data for a new Google service). By leveraging this information in the forecasting model, it is possible to improve the accuracy of predictions and mitigate the cold-start problem to some extent. For example, assume that in data center L_i, the historical data for attributes a_2 and a_{A-1} is not available but the future values of attribute a_{A-1} which is correlated with a_2 are available. Then, the forecasting task would be:

$$\begin{bmatrix} \hat{x}^{(1)}_{i,t+1} & \hat{x}^{(2)}_{i,t+1} & \dots & x^{(A-1)}_{i,t+1} & \hat{x}^{(A)}_{i,t+1} \\ \hat{x}^{(1)}_{i,t+2} & \hat{x}^{(2)}_{i,t+2} & \dots & x^{(A-1)}_{i,t+2} & \hat{x}^{(A)}_{i,t+2} \\ \dots & \dots & \dots & \dots & \\ \hat{x}^{(1)}_{i,t+H} & \hat{x}^{(2)}_{i,t+H} & \dots & x^{(A-1)}_{i,t+H} & \hat{x}^{(A)}_{i,t+H} \end{bmatrix} = F_\theta\Big(\begin{bmatrix} x^{(1)}_{i,t-U+1} & \emptyset & \dots & \emptyset & x^{(A)}_{i,t-U+1} \\ x^{(1)}_{i,t-U+2} & \emptyset & \dots & \emptyset & x^{(A)}_{i,t-U+2} \\ \dots & \dots & \dots & \dots & \\ x^{(1)}_{i,t} & \emptyset & \dots & \emptyset & x^{(A)}_{i,t} \\ - & - & \dots & x^{(A-1)}_{i,t+1} & - \\ \dots & \dots & \dots & \dots & \dots \\ - & - & - & x^{(A-1)}_{i,t+H} & - \end{bmatrix} \Big)$$

In this paper, our goal is to predict $(\hat{x}_{i,t}, \hat{x}_{i,t+1}, ..., \hat{x}_{i,t+H})$ as accurate as possible. More formally:

Problem 1 (cold-start forecasting) *Given N multivariate time series corresponds to L data centers with A attributes in T timestamps represented by* $\mathbf{X} \in R^{L \times T \times A}$*, along with a forecasting horizon of H, we aim to learn a function* F_θ *that predicts the future values of attributes such that:*

$$\text{argmin} \quad \sum_{h=1}^{H} \sum_{j=1}^{A} (\hat{x}^j_{i,t+h} - x^j_{i,t+h})^2. \tag{2}$$

4 Cold Causal Demand Forecasting Model

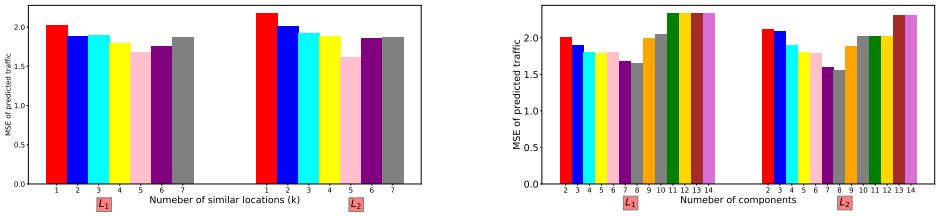
With the aim of addressing the cold-start problem in multivariate time series forecasting, we propose *Cold Causal Demand Forecasting (CDF-cold)* framework, which brings together causal inference and neural networks to improve forecasting accuracy for multivariate time series datasets suffering from cold-start problem. In this section, we provide a detailed description of the architecture of the CDF-cold framework. CDF-cold comprises two main components: 1) Causal Demand Forecasting, and 2) Cold-start Forecasting.

4.1 Causal Demand Forecasting (CDF)

The CDF component is designed to train a forecasting model for datasets with historical data for all variables. CDF consists of two sub-components:

- **Causal Component**. Existing time series forecasting models often assume correlations between a time series and its lags [1]. However, the future values of a time series can be influenced not only by its historical data but also by other variables in the dataset. Causal inference can play a crucial role in time series forecasting by identifying the causal relationships between variables in the dataset, improving our understanding of how different factors affect the time series over time, and helping us make more accurate predictions about future values. In our framework, we leverage a causal discovery algorithm to capture interdependencies between different variables and integrate the causal graph into the forecasting model. Structural Causal Models (SCMs) are graphical representations of cause-effect relationships between variables allowing for the identification of the effect of interest [33]. Causal graphs are Directed Acyclic Graph (DAG) representations of SCMs, where the direction of an edge determines the relationship between the variables. If the variable Y is the child of a variable X, then we say that Y is caused by X, or that X is the direct cause of Y.
- **Representation Learning Component**. The goal of this component is to learn a representation for each time series based on lags of the time series and variables causally impacting it. This component is a multilayer neural network model consisting of:
  - Graph Neural Network layer: For representation learning based on other attributes, we leverage *Graph Neural Networks (GNN)* whose effectiveness has been demonstrated in various machine learning tasks [52, 45, 38]. GNN is a deep learning approach for semi-supervised learning on graph-structured data which gets the matrix representation of the graph structure and the attribute matrix as the input and generates a new representation for each node based on the attributes of its neighbors. In our setting, for each data center L_i, We feed the causal graph G, extracted by the Causal Component, and the multivariate time series x_i into the GNN layer to obtain a new representation for each target variable based on the variables causally impacting it within the causal graph as:
  $$h_i = \sigma((x_i M)\mathbf{W_1}) \tag{3}$$
  where $M \in R^{A\times A}$ is the adjacency matrix corresponding to the causal graph G, $\mathbf{W_1} \in R^{T\times A\times D_g}$ is the weight matrix to be learned, D_g denotes the dimension of the new representation generated by GNN and σ stands for the ReLU activation function. By applying the GNN layer to the input multivariate time series and the causal model, an attribute's representation is informed by the information from its neighbors. However, we need not only information from the causally impacting attributes but also we need to process the information of each attribute over time. With this goal, we pass each attribute's new representation through a recurrent layer.
  - Long Short-Term Memory Networks (LSTM): LSTM is a Recurrent Neural Network (RNN) model with the capability of memorizing the important parts of the input sequence seen so far for the purpose of future use [19]. The LSTM layer enables the model to memorize historical information for each time series and generate a new representation based on the representation generated by the GNN layer. For a model with s LSTM layers, we have:
  $$\hat{h}_i = \sigma(...\sigma(\sigma(h_i\mathbf{W_2^1})\mathbf{W_2^2})...W_2^s) \tag{4}$$
  where $\mathbf{W_2^s} \in R^{A\times D_l}$ and D_l denotes the dimension of the new representation generated by the LSTM layer s.
  - Dense layer: The output of LSTM is passed to a densely connected neural network layer to generate the forecasting for H horizons as:
  $$h'_i = \sigma(\hat{h_i}\mathbf{W_3}) \tag{5}$$
  where $\mathbf{W_3} \in R^{A\times H}$ represents the learnable weights for the dense layer. The output of the dense layer shows the forecasting for A attributes in time steps $t+1, t+2, ..., t+H$.

  Since we assume that the future values for some attributes are available in advance, we take advantage of this information to improve the forecasting of other time series. Let $\mathbf{O_1}$ denote the set of attributes with known future values $x_{i,H}^{|\mathbf{O_1}|} \in R^{H\times|\mathbf{O_1}|}$ in data center L_i and let $\mathbf{O_2}$ represents all the attributes in A which are not in $\mathbf{O_1}$. We concatenate the forecasting of set $\mathbf{O_2}$ in h'_i with $x_{i,H}^{\mathbf{O_1}}$ and obtain a new vector $\tilde{h_i}$ as:
  $$\tilde{h_i} = concat(h_i'^{|\mathbf{O_2}|}, x_{i,H}^{|\mathbf{O_1}|}). \tag{6}$$



(a) CDF_eros (b) CDF_GMM

Figure 2: The impact of the number of similar data centers on the forecasting model error.

Then, we feed $\tilde{h_i}$ to a densely connected neural network layer to obtain the final forecasting:

$$\ddot{h}_i = \sigma(\tilde{h_i}\mathbf{W_4}), \tag{7}$$

where $\mathbf{W_4} \in R^{|\mathbf{O_2}|\times H}$ is the wight matrix to be learned. Fig. 1 demonstrates the network architecture of the CDF component.

4.2 Cold-start forecasting component

The utilization of the Representation Learning component enables the prediction of future values in time series datasets that have access to historical data. However, when faced with the cold-start problem, the model becomes insufficient in capturing the interdependencies between the target variable and variables without historical data. To address this issue, we propose a similarity-based approach that leverages forecasting models trained on the k most similar data centers with historical data. This component is comprised of three main steps as described below:

1. We use a similarity-based approach to find k most similar data centers to the target data center with the cold-start problem. A data center is considered a similar data center if it has enough historical data for the variables without historical data in the target data center.
2. We use the forecasting models trained on the k similar data centers to predict the future values of attributes in the target data center with no historical data. The input to each trained model would be the available future data for set $\mathbf{O_1}$ of the target data center $x_{i,H}^{|\mathbf{O_1}|}$.
3. We take the average of the predictions for target variables of the target data center predicted by the models trained on the k similar data centers.

We consider two different similarity-based approaches in our framework:

- Gaussian Mixture Model (GMM): GMM is a clustering technique that assumes a specific number of Gaussian distributions in the data, where each distribution represents a cluster [35]. By applying GMM to multiple multivariate time series datasets, we can group time series belonging to a single distribution. The parameters of the GMM model are estimated using the Expectation-Maximization algorithm based on maximum likelihood.
- Extended Frobenius norm (Eros): Eros is a Principal Component Analysis (PCA) based approach that measures the pairwise similarity between multiple multivariate time series datasets [48]. In this approach, the covariance matrices of different datasets are measured. Then, the similarity between eigenvectors weighted by eigenvalues of the covariance matrices quantifies the similarity between different datasets.

5 Experiments

In this section, we evaluate the performance of the baselines in multivariate time series forecasting. We first describe the dataset used in our experiments and then discuss our baselines and results.

5.1 Dataset

We collect the multivariate time series of 15 Google data centers. Each dataset comprises network traffic and machine usage information for the top 200 Google services in terms of network traffic, as well as the overall network traffic for each data center, over a period of 533 days.

5.2 Baselines

We compare the performance of our framework with different forecasting models:

- LSTM: For each data center, we train a forecasting model comprising an LSTM layer and a dense layer using the multivariate time series dataset of the data center.
- GNN+LSTM: This approach is similar to the CDF model but we assume that the causal effect of each variable on all other variables is equivalent to 1 ($\forall m_{i,j} \in M$, $m_{i,j} = 1$).
- CDF: This is our proposed forecasting framework represented in Fig. 1.
- CDF_GMM: This method is considered as a variant of the Cold-CDF framework. In this model, GMM is applied in the cold-start forecasting component of the CDF-cold framework to find the most similar data centers to the target data center with the cold-start problem.
- CDF_GMM_sd: This approach is a variant of CDF_GMM method. In this technique, GMM is used to identify the most similar data centers to the target data center. Then, the CDF models trained on similar data centers are used to forecast the future values of the time series in the target data center. We iteratively remove the forecasts which are out of the range of standard deviation of the remaining forecasts. Finally, we take the average of the remaining forecasts as the predicted values for the target data center.
- CDF_eros: In this approach, the Eros method is used to measure the similarities between different multivariate time series in the cold-start forecasting component.
- CDF_virtual: In this approach, GMM is used to identify k most similar data centers to the target data center with the cold-start problem. Then, a virtual data center is created by measuring the pointwise average of the time series of similar data centers. Finally, a CDF model is trained using the virtual data center and exploited to forecast in the target center.
- CDF_virtual_mn: This model is a variant of CDF_virtual. In this approach, a virtual data center is created by measuring the pointwise average of the time series of similar data centers weighted by the Manhattan distance of the target data center and each similar data center.

5.3 Experimental Setup

To smooth time series data over outliers and short-term fluctuations, we use the rolling median technique in which the attribute values of a sliding time window are replaced with the median of the values in that window. We set the window size to 7 and utilize first-order differencing ($x^j_{i,t} \leftarrow x^j_{i,t} - x^j_{i,t-1}$) to convert non-stationary datasets to stationery. In non-stationary time series datasets, the statistical properties of the dataset (e.g., mean and variance) change in time. To standardize the dataset, we use Z-score normalization in which attributes are rescaled to ensure the mean and the standard deviation are 0 and 1, respectively.

We set the observable past window size $U = 10$ and horizon $H \in \{1, 10\}$. We use the Mean Squares Error (MSE) loss function and the RMSProp optimizer to optimize the parameters of our model. For causal discovery, we exploit VARLiNGAM [21] which is an extension of the LiNGAM [39] model to time series datasets. VARLiNGAM enables analyzing both lagged and contemporaneous (instantaneous) causal relations in multivariate time series datasets.

To evaluate the effectiveness of various forecasting models in datasets with the cold-start problem, we remove the historical data for ten of the Google services with high network traffic until t=400 in each data center. We then predict the total network traffic beyond this time step (H $\in \{10, 20\}$) when the removed service is added to the data center. Each time, we select one data center as the dataset with the cold-start problem, we consider the other 14 data centers as potentially similar data centers. To assess the performance of different models, we follow existing literature [27, 46] and report the mean absolute error (MAE), mean square error (MSE), and mean absolute percentage error (MAPE) of the total network traffic in each data center.

5.4 Results

Sensitivity to the number of similar data centers: In this experiment, our objective is to explore how the number of similar data centers affects the performance of different cold-start forecasting component variants. As shown in Fig. 2, our findings demonstrate that the model trained exclusively

Table 1: Comparison between the forecasting of different methods for $\mathbf{H} \in \{\mathbf{10}, \mathbf{20}\}$ and $\mathbf{U} = \mathbf{12}$.

		H=10			H=20		
Data center	Metric	**LSTM**	**LSTM+GNN**	**CDF**	**LSTM**	**LSTM+GNN**	**CDF**
L_1	MSE	0.088	0.039	**0.012**	0.21	0.03	**0.001**
	MAE	0.2	0.165	**0.08**	0.4	0.1	**0.03**
	MAPE	0.91	0.7	**0.52**	0.18	0.1	**0.07**
L_2	MSE	3.88	3.54	**3.1**	0.85	0.41	**0.32**
	MAE	1.43	1.35	**1.27**	0.89	0.63	**0.57**
	MAPE	3.34	2.41	**2.27**	1.19	0.94	**0.66**
L_3	MSE	9.6	8.69	**7.65**	0.35	0.13	**0.024**
	MAE	2.19	2.1	**1.87**	0.47	0.16	**0.95**
	MAPE	2.08	1.48	**1.21**	4.23	1.00	**0.95**
L_4	MSE	0.28	0.18	**0.11**	0.26	0.15	**0.008**
	MAE	0.59	0.47	**0.27**	0.41	0.18	**0.079**
	MAPE	4.68	2.67	**2.23**	16.49	8.87	**3.89**

Table 2: Comparison between the performance of different methods in mitigating cold-start forecasting problem for H=10 and U=12. The best results are highlighted in bold.

Data center	Metric	**LSTM**	**LSTM+GNN**	**CDF**	**CDF_eros**	**CDF_GMM**	**CDF_GMM_sd**	**CDF_virtual**	**CDF_virtual_mn**
L_1	MSE	2.14	1.98	1.85	1.76	1.64	**1.41**	1.69	1.66
	MAE	1.282	1.12	0.99	0.98	0.95	**0.9**	0.98	0.98
	MAPE	1.63	1.35	1.24	1.07	0.98	**0.91**	1.4	1.18
L_2	MSE	2.34	1.98	1.75	1.61	1.55	**1.40**	1.64	1.62
	MAE	0.99	0.91	0.86	0.82	0.79	**0.77**	0.85	0.82
	MAPE	2.99	2.21	2.07	1.86	1.71	**1.63**	1.9	1.81
L_3	MSE	3.11	2.98	2.71	2.68	2.53	**2.31**	2.63	2.56
	MAE	1.22	1.14	1.08	1.04	1.01	**1.01**	1.04	1.02
	MAPE	1.87	1.51	1.49	1.47	1.19	**1.12**	1.59	1.51
L_4	MSE	0.42	0.28	0.2	0.17	0.12	**0.09**	0.18	0.16
	MAE	0.59	0.41	0.31	0.28	0.25	**0.24**	0.29	0.27
	MAPE	10.67	2.61	2.12	1.88	1.51	**1.50**	1.83	1.78

on the most similar data center does not necessarily yield the most precise forecasting for data centers that encounter the cold-start problem. In the CDF_eros method, averaging the forecast of the models trained on the first five most similar data centers produces the least forecasting error. Considering the CDF_GMM method, we observe that partitioning the dataset into 2 or 3 dense clusters or sparse clusters with 12, 13, and 14 partitions leads to a significantly higher forecasting error. Conversely, configuring the number of components to 7 or 8 results in the lowest prediction error.

Evaluating the forecasting performance: In this experiment, the performance of three different models in total network traffic forecasting was compared. Table 1 presents the MSE, MAE, and MAPE of total network traffic predicted by three different methods in four different Google data centers. The results for more datsets can be found in Appendix. The results show that applying GNN can significantly improve the accuracy of forecasting compared to LTSM in all data centers. Compared to LSTM+GNN, CDF decreases the error from significantly. To study the ability to forecast longer horizons, we increase the forecasting horizon from $H = 10$ to $H = 20$. Similar to the results when $H = 10$, GMM outperforms the LSTM model. Furthermore, we observe that CDF with a causal component consistently outperforms LSTM+GNN in all datasets.

Cold-start forecasting evaluation: To evaluate different methods for mitigating the cold-start forecasting, we MSE, MAE, and MAPE of total network traffic prediction using the forecasting methods. As depicted in Table 2, the methods that lack a cold-start forecasting component (LSTM, LSTM+GNN, and CDF) exhibit higher estimation error compared to the methods that include a cold-start forecasting component. This can be attributed to the fact that LSTM cannot, LSTM+GMM, and CDF cannot learn the interdependence between variables and the total network traffic without historical data for some variables. However, we observed that CDF outperforms LSTM and LSTM+GNN in all datasets which is consistent with the results reported in Table. 1. Among the Cold-CDF variants, cluster-based methods (i.e., CDF_GMM, and CDF_GMM_sd) exhibit the least forecasting error. Compared to CDF, the CDF_GMM_sd approach improves the MSE in all datasets. The findings for the other two metrics (MAE and APE) are consistent with these results. CDF_GMM and CDF_GMM_sd also demonstrate the lowest values for MAE and MAPE.

6 Conclusion

In this paper, we present the Cold Causal Demand Forecasting framework, a novel approach designed to address cold-start forecasting in multivariate time series data where some variables lack historical information. Our framework leverages causal discovery algorithms to uncover cause-and-effect relationships among interdependent variables. These discovered relationships are then integrated into a neural network model, resulting in improved forecasting accuracy. We propose a similarity-based approach to tackle the cold-start problem. Our comprehensive evaluation of 15 Google multivariate

time series datasets reveals the superior performance of our framework in comparison to existing baseline methods. One potential future direction is to integrate transformers and causal inference to propose a model for datasets with no historical data for all variables.

References

[1] Farid Khalil Arya and Lan Zhang. Time series analysis of water quality parameters at stillaguamish river using order series method. *Stochastic environmental research and risk assessment*, 29:227–239, 2015.

[2] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[3] C Bui, N Pham, A Vo, A Tran, A Nguyen, and T Le. Time series forecasting for healthcare diagnosis and prognostics with the focus on cardiovascular diseases. In *6th International Conference on the Development of Biomedical Engineering in Vietnam (BME6) 6*, pages 809–818. Springer, 2018.

[4] Jiasheng Cao and Jinghan Wang. Stock price forecasting model based on modified convolution neural network and financial time series analysis. *International Journal of Communication Systems*, 32(12):e3987, 2019.

[5] Nancy Cartwright. *Hunting Causes and Using Them: Approaches in Philosophy and Economics*. Cambridge University Press, 2007.

[6] Yonghong Chen, Govindan Rangarajan, Jianfeng Feng, and Mingzhou Ding. Analyzing multiple nonlinear time series with extended granger causality. *Physics letters A*, 324(1):26–35, 2004.

[7] Dawei Cheng, Fangzhou Yang, Sheng Xiang, and Jin Liu. Financial time series forecasting with multi-modality graph neural network. *Pattern Recognition*, 121:108218, 2022.

[8] Razvan-Gabriel Cirstea, Bin Yang, Chenjuan Guo, Tung Kieu, and Shirui Pan. Towards spatio-temporal aware traffic time series forecasting. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 2900–2913. IEEE, 2022.

[9] Mostafa Dastorani, Mohammad Mirzavand, Mohammad Taghi Dastorani, and Seyyed Javad Sadatinejad. Comparative study among different time series models applied to monthly rainfall forecasting in semi-arid climate condition. *Natural Hazards*, 81:1811–1827, 2016.

[10] Rodrigo de Medrano and Jose L Aznarte. A spatio-temporal attention-based spot-forecasting framework for urban traffic prediction. *Applied Soft Computing*, 96:106615, 2020.

[11] Shengdong Du, Tianrui Li, Yan Yang, and Shi-Jinn Horng. Multivariate time series forecasting via attention-based encoder–decoder framework. *Neurocomputing*, 388:269–279, 2020.

[12] Michael Eichler. Granger causality and path diagrams for multivariate time series. *Journal of Econometrics*, 137(2):334–353, 2007.

[13] John R Freeman. Granger causality and the times series analysis of political relationships. *American Journal of Political Science*, pages 327–358, 1983.

[14] Michael Ghil, MR Allen, MD Dettinger, K Ide, D Kondrashov, ME Mann, Andrew W Robertson, A Saunders, Y Tian, Fa Varadi, et al. Advanced spectral methods for climatic time series. *Reviews of geophysics*, 40(1):3–1, 2002.

[15] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.

[16] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.

[17] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1):388–427, 2021.

[18] Siu Lau Ho and Min Xie. The use of arima models for reliability forecasting and analysis. *Computers & industrial engineering*, 35(1-2):213–216, 1998.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[20] Biwei Huang, Kun Zhang, Mingming Gong, and Clark Glymour. Causal discovery and forecasting in nonstationary environments with state-space models. In *International conference on machine learning*, pages 2901–2910. PMLR, 2019.

[21] Aapo Hyvärinen, Kun Zhang, Shohei Shimizu, and Patrik O Hoyer. Estimation of a structural vector autoregression model using non-gaussianity. *Journal of Machine Learning Research*, 11(5), 2010.

[22] Shruti Kaushik, Abhinav Choudhury, Pankaj Kumar Sheron, Nataraj Dasgupta, Sayee Natarajan, Larry A Pickett, and Varun Dutt. Ai in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. *Frontiers in big data*, 3:4, 2020.

[23] Kyoung-jae Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2):307–319, 2003.

[24] Gebhard Kirchgässner, Jürgen Wolters, Uwe Hassler, Gebhard Kirchgässner, Jürgen Wolters, and Uwe Hassler. Granger causality. *Introduction to modern Time Series analysis*, pages 95–125, 2013.

[25] Bjoern Krollner, Bruce J Vanstone, Gavin R Finnie, et al. Financial time series forecasting with machine learning techniques: a survey. In *ESANN*, 2010.

[26] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. 2017.

[27] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. 2017.

[28] Youru Li, Zhenfeng Zhu, Deqiang Kong, Hua Han, and Yao Zhao. Ea-lstm: Evolutionary attention-based lstm for time series prediction. *Knowledge-Based Systems*, 181:104785, 2019.

[29] Rishabh Madan and Partha Sarathi Mangipudi. Predicting computer network traffic: A time series forecasting approach using dwt, arima and rnn. In *2018 Eleventh International Conference on Contemporary Computing (IC3)*, pages 1–5, 2018.

[30] Francisco Martínez-Álvarez, Alicia Troncoso, Gualberto Asencio-Cortés, and José C Riquelme. A survey on data mining techniques applied to electricity-related time series forecasting. *Energies*, 8(11):13162–13193, 2015.

[31] Prapanna Mondal, Labani Shit, and Saptarsi Goswami. Study of effectiveness of time series modeling (arima) in forecasting stock prices. *International Journal of Computer Science, Engineering and Applications*, 4(2):13, 2014.

[32] CK Moorthy and BG Ratcliffe. Short term traffic forecasting using time series methods. *Transportation planning and technology*, 12(1):45–56, 1988.

[33] J Pearl. *Causality*. Cambridge Univ Press, 2009.

[34] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31, 2018.

[35] Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.

[36] Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *INTERSPEECH*, 2014.

[37] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[38] Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. In *International Conference on Learning Representations*, 2021.

[39] Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006.

[40] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pages 1394–1401. IEEE, 2018.

[41] Ireneous N Soyiri and Daniel D Reidpath. An overview of health forecasting. *Environmental health and preventive medicine*, 18(1):1–9, 2013.

[42] Yuanrong Wang and Tomaso Aste. Sparsification and filtering for spatial-temporal gnn in multivariate time-series. *arXiv preprint arXiv:2203.03991*, 2022.

[43] Yuyang Wang, Alex Smola, Danielle Maddix, Jan Gasthaus, Dean Foster, and Tim Januschowski. Deep factors for forecasting. In *International conference on machine learning*, pages 6607–6617. PMLR, 2019.

[44] Yue Wu, José Miguel Hernández-Lobato, and Ghahramani Zoubin. Dynamic covariance models for multivariate financial time series. In *International Conference on Machine Learning*, pages 558–566. PMLR, 2013.

[45] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 753–763, 2020.

[46] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 1907–1913, 2019.

[47] Haoyan Xu, Yida Huang, Ziheng Duan, Jie Feng, and Pengyu Song. Multivariate time series forecasting based on causal inference with transfer entropy and graph neural network. *arXiv preprint arXiv:2005.01185*, 2020.

[48] Kiyoung Yang and Cyrus Shahabi. A pca-based similarity measure for multivariate time series. In *Proceedings of the 2nd ACM international workshop on Multimedia databases*, pages 65–74, 2004.

[49] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3634–3640, 2018.

[50] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3634–3640, 2018.

[51] Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. Lstm network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2):68–75, 2017.

[52] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

Table 3: Comparison between the forecasting of different methods for $\mathbf{H} \in \{\mathbf{10}, \mathbf{20}\}$ and $\mathbf{U} = \mathbf{12}$.

		H=10			H=20		
Data center	Metric	**LSTM**	**LSTM+GNN**	**CDF**	**LSTM**	**LSTM+GNN**	**CDF**
L_5	MSE	1.84	1.69	**1.46**	6.02	3.28	**2.82**
	MAE	0.99	0.96	**0.88**	1.52	1.26	**1.00**
	MAPE	2.04	1.05	**1.10**	0.68	0.60	**0.35**
L_6	MSE	1.38	1.21	**1.12**	4.2	0.028	**0.004**
	MAE	0.79	0.76	**0.69**	0.46	0.169	**0.058**
	MAPE	2.56	2.16	**1.65**	2.05	0.72	**0.25**
L_7	MSE	1.48	1.18	**0.98**	4.21	3.16	**2.82**
	MAE	0.95	0.83	**0.77**	2.03	1.72	**1.68**
	MAPE	2.21	1.78	**1.30**	1.09	0.92	**0.6**
L_8	MSE	1.33	1.02	**0.82**	0.21	0.01	**0.001**
	MAE	0.71	0.64	**0.57**	0.42	0.08	**0.017**
	MAPE	2.00	1.5	**1.17**	3.53	0.33	**0.87**
L_9	MSE	1.17	0.73	**0.47**	0.58	0.21	**0.062**
	MAE	0.95	0.68	**0.57**	0.75	0.24	**0.11**
	MAPE	3.34	2.42	**1.60**	1.71	0.54	**0.38**
L_{10}	MSE	1.88	1.69	**1.34**	11.91	9.04	**7.97**
	MAE	0.85	0.65	**0.44**	2.94	2.63	**2.3**
	MAPE	2.17	1.78	**1.39**	1.29	1.05	**0.86**
L_{11}	MSE	7.71	6.88	**6.37**	3.41	1.93	**1.31**
	MAE	2.5	2.36	**2.26**	1.61	1.26	**1.07**
	MAPE	1.07	0.99	**0.94**	1.04	0.99	**0.75**
L_{12}	MSE	0.93	0.74	**0.6**	2.5	1.005	**1.31**
	MAE	0.73	0.66	**0.58**	1.53	1.05	**0.84**
	MAPE	2.63	2.28	**1.39**	1.19	0.98	**0.78**
L_{13}	MSE	2.05	1.69	**1.53**	10.05	6.89	**5.98**
	MAE	1.01	0.94	**0.82**	3.15	2.61	**2.13**
	MAPE	2.41	2.19	**1.08**	1.09	0.98	**0.81**
L_{14}	MSE	2.275	1.14	**0.94**	0.055	0.016	**0.014**
	MAE	1.12	0.88	**0.79**	0.22	0.1	**0.08**
	MAPE	2.76	1.14	**1.02**	2.65	0.68	**0.47**
L_{15}	MSE	3.83	3.78	**3.32**	0.75	0.24	**0.14**
	MAE	1.66	1.56	**1.49**	0.83	0.49	**0.39**
	MAPE	1.92	1.32	**0.93**	1.27	0.75	**0.55**

A Appendix

A.1 Hyperparameter Tuning

We vary hyper-parameters for each baseline method and each dataset to achieve their best performance on this task. We split each dataset into 80% training, 10% validation, and 10% test datasets. To train LSTM, GNN and CDF models, we search the learning rate in $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$, the number of epochs in $\{10, 20, 30, 50, 70, 100\}$ and the batch size in $\{16, 32, 64, 128\}$. The number of hidden units is chosen from $\{10, 20, 100, 200, 300\}$ and the number of hidden layers is set from $\{1, 2, 3\}$.

A.2 Results

This section contains the results for comparison between different forecasting models in predicting the future values in datasets with historical data and datasets with a cold-start problem.

Table 4: Comparison between the performance of different methods in mitigating cold-start forecasting problem for H=10 and U=12. The best results are highlighted in bold. CDF_GMM_sd outperforms other methods in forecasting for all data centers.

Data center	Metric	**LSTM**	**LSTM+GNN**	**CDF**	**CDF_eros**	**CDF_GMM**	**CDF_GMM_sd**	**CDF_virtual**	**CDF_virtual_mn**
L_5	MSE	1.52	1.32	1.08	0.91	0.86	**0.73**	0.9	0.87
	MAE	0.96	0.81	0.69	0.68	0.66	**0.56**	0.72	0.69
	MAPE	2.31	1.91	1.54	1.32	1.21	**1.12**	1.78	1.56
L_6	MSE	0.89	0.78	0.67	0.61	0.5	**0.37**	0.6	0.53
	MAE	0.76	0.61	0.53	0.49	0.42	**0.41**	0.5	0.49
	MAPE	3.12	1.78	1.52	1.48	1.30	**1.26**	1.37	1.31
L_7	MSE	3.24	2.97	2.51	2.42	2.33	**2.18**	2.49	2.38
	MAE	1.43	1.21	1.09	1.03	1.01	**0.99**	1.02	1.00
	MAPE	1.65	1.35	1.21	1.12	1.02	**0.95**	1.2	1.14
L_8	MSE	1.29	0.98	0.77	0.59	.43	**0.31**	0.58	0.55
	MAE	0.68	0.54	0.49	0.48	0.43	**0.40**	0.53	0.48
	MAPE	2.92	1.96	1.84	1.52	1.45	**1.38**	2.69	2.13
L_9	MSE	0.99	0.88	0.71	0.65	0.51	**0.40**	0.67	0.64
	MAE	0.83	0.71	0.59	0.53	0.52	**0.51**	0.53	0.52
	MAPE	2.77	2.32	1.98	1.63	1.39	**1.38**	2.10	2.00
L_{10}	MSE	0.86	0.75	0.61	0.53	0.41	**0.34**	0.58	0.54
	MAE	0.69	0.58	0.46	0.4	0.38	**0.37**	0.42	0.40
	MAPE	3.71	2.78	2.43	2.29	2.15	**1.69**	2.75	2.21
L_{11}	MSE	3.67	2.98	2.74	2.69	2.42	**2.11**	2.72	2.69
	MAE	1.78	1.37	1.33	1.28	1.19	**1.02**	1.33	1.29
	MAPE	1.83	1.54	1.50	1.21	1.07	**1.00**	1.16	1.08
L_{12}	MSE	1.93	1.63	1.58	1.53	1.42	**1.19**	1.59	1.51
	MAE	1.02	0.96	0.86	0.8	0.76	**0.71**	0.9	0.78
	MAPE	3.72	2.89	2.54	1.47	1.38	**1.21**	1.65	1.40
L_{13}	MSE	2.53	2.02	1.81	1.64	1.51	**1.28**	1.98	1.78
	MAE	1.16	1.03	0.99	0.98	0.95	**0.89**	1.04	1.01
	MAPE	6.52	5.67	4.22	3.90	3.62	**2.65**	10.64	7.04
L_{14}	MSE	0.98	0.89	0.76	0.71	0.61	**0.50**	0.87	0.7
	MAE	0.83	0.75	0.96	0.84	0.68	**0.68**	0.75	0.69
	MAPE	1.84	1.70	1.56	1.22	1.07	**1.02**	1.8	1.65
L_{15}	MSE	4.21	3.93	3.76	3.21	3.05	**2.82**	3.76	3.43
	MAE	1.52	1.43	1.31	1.3	1.24	**1.21**	1.37	1.27
	MAPE	1.62	1.58	1.42	1.21	1.09	**1.01**	2.44	1.16