

A Novel Deep Learning Model For Hotel Demand and Revenue Prediction amid COVID-19

Ashkan Farhangi
University of Central Florida
ashkan.farhangi@ucf.edu

Arthur Huang
University of Central Florida
arthur.huang@ucf.edu

Zhishan Guo
University of Central Florida
zsguo@ucf.edu

Abstract

The COVID-19 pandemic has cast a substantial impact on the tourism and hospitality sector. Public policies such as travel restrictions and stay-at-home orders had significantly affected tourist activities and service businesses' operations and profitability. It is essential to develop interpretable forecasting models to support managerial and organizational decision-making. We developed DemandNet, a novel deep learning framework for predicting time series data under the influence of the COVID-19 pandemic. The DemandNet framework has the following unique characteristics. First, it selects the top static and dynamic features embedded in the time series data. Second, it includes a nonlinear model which can provide interpretable insight into the previously seen data. Third, a novel prediction model is developed to leverage the above characteristics to make robust long-term forecasts. We evaluated DemandNet using daily hotel demand and revenue data from eight cities in the US between 2013 and 2020. Our findings reveal that DemandNet outperforms the state-of-art models and can accurately predict the effect of the COVID-19 pandemic on hotel demand and revenue.

1. Introduction

The COVID-19 pandemic has substantially disrupted global economic activities. The stay-at-home order and business closure policy intended to curb the COVID-19 transmission have taken an unintended toll on the economy [1]. As large volumes of business data become available, it is critical to develop machine learning models and tools to shed light on the impact of COVID-19 and policies on demand and revenue and unpack the future trends [2, 3]. Such machine learning models should offer both high accuracy and interpretability to assist managers in making informed decisions about new business strategies and market recovery. [4] The fine-grained business data amid crises

are high-dimensional time series observations with great complexity. They are difficult to be sufficiently modeled using traditional time series models due to their dynamic nature and not frequently seen uncertainties [1, 5, 6]. Furthermore, current deep learning models contain complex and nonlinear parameters that are often uninterpretable [7]. In addition, they rarely provide uncertainty measurement in their predictions, which becomes a challenge when the data are influenced by external shocks such as the COVID-19 pandemic [8].

Given that the COVID-19 pandemic has caused a sudden change in the data trend and seasonality, it is vital to develop a new deep learning approach for modeling real-time data under the influence of external shocks. The new approach should be interpretable and reliable and would uncover the effects (e.g., correlation and strengths) of exogenous variables on the interested variables.

To this end, we propose DemandNet, a novel deep learning framework that can capture the essential features in time series data and achieve interpretability. Specifically, a carefully designed feature selection mechanism is proposed to filter appropriate information from the prior pandemic data. Furthermore, we develop a multilayer neural network to derive the nonlinear relation of the selected features to specific economic responses (business and individuals). The new approach is tested on daily hotel demand and revenue data in eight major cities in the US between 2013 and 2020. The results show that the model is able to learn the impact of COVID-19 from the training set and make accurate predictions of hotel demand and revenue on unseen (not trained) time series data.

Contributions. This study includes the following contributions:

- We design a feature selection mechanism to select the top static and dynamic features of a time series. Such capability also enhances the DemandNet's ability to capture complex critical features.

- We design a multilayer neural network to derive the nonlinear relation of the selected features to the predictor. Such nonlinear relation can then provide interpretable insights about the past and be an aid in future prediction.
- We developed a novel multi-step time series prediction model that leverages a dynamic dropout optimization mechanism to minimize the prediction uncertainties and provide an optimal level of confidence in forecasts.
- The framework is capable of predicting newly added time series data without any previous training. Such abilities are verified with extensive experiments.

Organization. The rest of the paper is organized as follows. Section 2 formulates the demand prediction problem. Section 3 proposes DemandNet as a prediction method with dropout optimization for ensuring reliable uncertainty. Section 4 presents the datasets and metrics, discusses the experimental results, and conducts an ablation study. Section 5 summarizes the related works and Section 6 concludes this paper.

2. Problem Formulation

In this work, we are interested in the task of time series prediction under the presence of COVID-19 pandemic. Mathematically, given a dataset with K univariate time series $\mathbf{D} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \dots, \mathbf{X}^{(K)}\}$, we use $\mathbf{X}^{(k)} = \{\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, \dots, \mathbf{x}_T^{(k)}\}$ to denote a period of data with length T where $\mathbf{X}^{(k)} \in R^T$.

For every observation in $\mathbf{X}^{(k)}$, the corresponding covariates are denoted as $\mathbf{V}^{(k)} = \{\mathbf{v}_1^{(k)}, \mathbf{v}_2^{(k)}, \dots, \mathbf{v}_T^{(k)}\}$. The covariates include up to M exogenous variables (e.g., COVID-19 cases, mobility), where $\mathbf{v}_t^{(k)} \in R^M$. As an example, the level of state policy at time t , can be shown as $\mathbf{v}_t^{(k)} \in R$; otherwise, $\mathbf{v}_t^{(k)} = 0$ indicates that there are no state policy at time t . Hence, given the previous τ observations $\widehat{\mathbf{X}}_{t-\tau+1:t}^{(k)} = \{(\mathbf{x}_{t-\tau+1}^{(k)}, \mathbf{v}_{t-\tau+1}^{(k)}), (\mathbf{x}_{t-\tau+2}^{(k)}, \mathbf{v}_{t-\tau+2}^{(k)}), \dots, (\mathbf{x}_t^{(k)}, \mathbf{v}_t^{(k)})\}$, the objective is predict till the H -th horizon time step $\mathbf{Y} = \mathbf{x}_{t+1:t+H}^{(k)}$. To this end, the optimization problem of DemandNet is defined as below:

$$\Phi^* = \operatorname{argmin}_{\Phi} \mathcal{L}(\mathcal{F}_{\Phi}(\widehat{\mathbf{X}}), \mathbf{Y}) \quad (1)$$

where Φ are the parameters of the non-linear function $\mathcal{F}(\mathbf{X})$ and \mathcal{L} is the loss function.

3. DemandNet Framework

We propose the DemandNet framework to model and predict the effect of the COVID-19 pandemic on the target times series as shown in Figure 1. There are three main components in DemandNet: First, DemandNet uses a feature selection mechanism to select the top correlated features (see Section 3.1). Second, the nonlinear modeling of hotel sales is derived based on anti-contagion policy (see Section 3.2). Third, the prediction model uses the selected features and the derived nonlinear model to provide robust and reliable multi-step predictions (see Section 3.3).

3.1. Feature Selection

Many frameworks designed for pandemic time series data do not incorporate top important features about their subjects [9]. Research shows that a set of important features can improve the learning process [10, 11]. However, a great number of features can lead to the curse of dimensionality, which limits the model's ability to incorporate essential features. To resolve this issue, we develop a feature selection mechanism that retains the important features.

With the current advances in feature extraction strategies, deriving a complete set of information from a pandemic and its impact on hotel demand and revenue is challenging. Often the relationship between the targeted variable and exogenous variables is unknown. It also is difficult to assume that all information is essential and relevant [7]. Therefore, we need a specific design for feature selection that filters the appropriate related information. To achieve this objective, we develop a two-step feature selection mechanism to select each time series's top static and dynamic features. The detailed procedure includes two steps:

First, our feature selection mechanism adopts Spearman rank correlation to filter the non-monotonic static features. Compared with Pearson correlation that evaluates the linear relationship of features, Spearman rank correlation provides the monotonic relation of features that tend to change together but not at a constant (i.e., linear) rate.

We start by deriving the correlation between every two sets of static features $a_{1:n}$ and $b_{1:n}$, each with n number of observations. Both pairs of features will then be ranked in ascending order based on their raw value. For identical values, we assign a rank equal to the average of their positions in the ascending order. To this end, the correlation coefficient r_S of each pair can

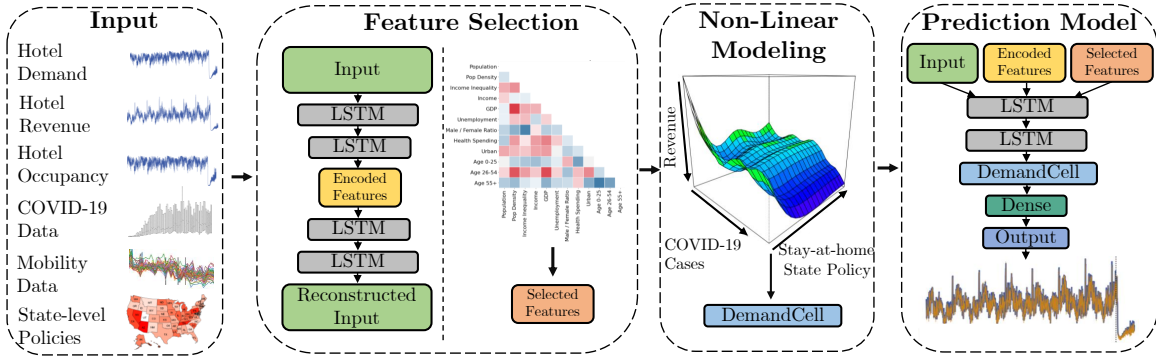


Figure 1. DemandNet consists of three main components: (1) the feature selection mechanism which provides the top static and dynamic features; (2) a nonlinear model which provides interpretability and captures the effect of the selected features; (3) a prediction model which reports the estimated forecast.

be calculated as:

$$r_S = \frac{\sum_{i=1}^n \{(a_i - \bar{a}) \cdot (b_i - \bar{b})\}}{\sqrt{\sum_{i=1}^n (a_i - \bar{a})^2} \cdot \sqrt{\sum_{i=1}^n (b_i - \bar{b})^2}}, \quad (2)$$

where both \bar{a} and \bar{b} are the average ranks. The coefficient can yield a value $-1 \leq r_S \leq 1$, which can be interpreted as having a minimum association ($r_S = 0$) or as a perfect monotonic relationship ($r_S = \pm 1$). Then, the feature selection mechanism removes the weak to moderate correlated features based on a region suggested by [12] (i.e., $-0.3 < r_S < 0.3$).

The exemplary Spearman Rank Correlations are shown in Figure 2, where it illustrates a heatmap of correlations from a set of static state-specific data. Such a correlational heatmap can be used to develop a profile for each time series and further improve the prediction accuracy as the model has access to important state-level economic, social, and health data [7]. Moreover, such correlations enable DemandNet to use highly correlated features for developing profiles for location-specific time series data. To provide statistical inference, entropy-based correlations can also be a good candidate if there we are given greater sample size.

Second, complex time series data often contain critical features during the presence of external shocks that are hidden even by the most advanced methods. To resolve this issue, we stack Long-Short Term Memory (LSTM) layers to form serve as a Stacked Autoencoder.

Specifically, the LSTM layers encode the time series into its top essential dynamic features (See Figure 1). The input signal is reconstructed at the output through an intermediate layer with a reduced amount of neurons. The middle (smallest) layer is chosen as a dense layer to hold the essential features needed for a proper reconstruction. Intuitively, the model retains

the deep and abstract features in the dense layer so that the reconstruction would be as close as possible to the original input. Note that we stack LSTMs layers to capture complex abstract information that are difficult to gather by shallower layers.

The encoder maps the input $\hat{\mathbf{x}} \in \mathbb{R}^d$ to the encoded features $\mathbf{u} \in \mathbb{R}^h$ in the encoding step where h is the number of neurons in the encoded features layer. Ideally, the number of neurons in the encoded feature (\mathbf{u}) should be chosen smaller than the stacked LSTM layers, so the network is compelled to keep only the top important abstract information. To this end, the decoder maps the encoded features to a reconstructed $\mathbf{z} \in \mathbb{R}^d$ in the decoding step.

We initialize the weights of the encoder and decoder randomly. Mathematically, the encoder \mathcal{B} and decoder \mathcal{D} are defined as:

$$\mathbf{U} = \mathcal{B}(w_u \hat{\mathbf{X}} + b_u) \quad (3)$$

$$\mathbf{Z} = \mathcal{D}(w_z \mathbf{U} + b_z) \quad (4)$$

where w_u and w_z denote the encoder and decoder weights. Coupled with b_u , and b_z that denote the bias of hidden and output units.

Then, we train the encoder-decoder network to derive the optimized parameters so that the error between $\hat{\mathbf{x}}$ and \mathbf{z} gets minimized, i.e.,

$$\underset{w_x, w_z, b_x, b_z}{\operatorname{argmin}} \operatorname{error}(\hat{\mathbf{X}}, \mathbf{Z}). \quad (5)$$

The training process is an iterative update of the parameters w and b , an update by which the error between the input e series and the reconstructed one at the network's output is steadily reduced until it is below a preset threshold. An effective training results in a decreased error, as displayed in Eq. (5), which ensures

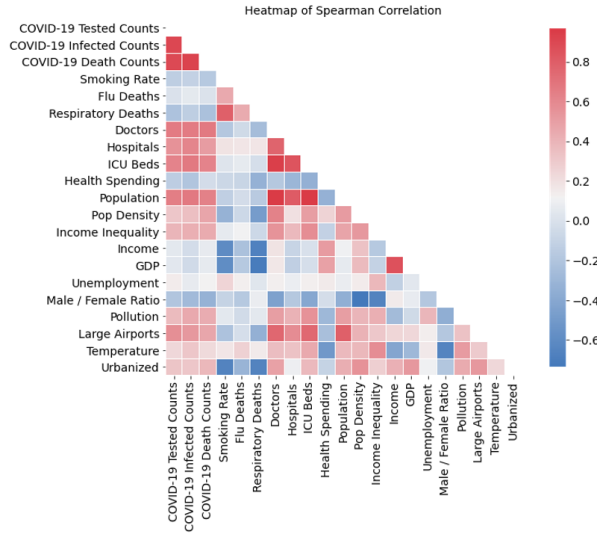


Figure 2. Heatmap of Spearman rank correlation for state-level static features. The feature selection mechanism removes the weakly correlated features.

the proper internal features are present as the encoded features \mathbf{u} .

Thus, the two-step feature selection procedure provides DemandNet with top features that can be used in the nonlinear model and multi-step time series prediction methods discussed in the next sections.

3.2. Nonlinear Modeling

Based on the top features selected in the previous section, we can now develop nonlinear models to discover the correlation of these features to the target data. Specifically, we are using a multilayer neural network to serve as a nonlinear model. Other than the interpretability, the nonlinear model can then be used as a control mechanism to improve the robustness of multi-step forecasts.

The multilayer NN is composed of an input layer (\mathbf{x}), hidden layer (\mathbf{h}), and an output layer (\mathbf{y}), where the hidden layer j can be represented as follows:

$$\mathbf{h}_j = \mathcal{G}(w_j \hat{\mathbf{x}} + b_j), \quad (6)$$

where w_j, b_j, u_j denote the weight, bias, and output of the j th hidden layer respectfully. Moreover, \mathcal{G} denotes a nonlinear function (e.g., sigmoid).

The modeling process starts by assigning random weights and biases for the hidden layers. Next, we feed the network our normalized data through the input layer, which then gets delivered to the hidden layers. Note that the network should be designed to avoid extreme values of weights to optimize the fit. Failure to do so will result

in predictions that will be unstable for extrapolation. Hence, we use a penalty function to obtain a more stable and smooth fit. To this end, the weights and biases of the hidden layers are chosen to minimize the following loss function:

$$L(y, y^*, w) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^m w_j^2, \quad (7)$$

where y is the predicted output, y^* is the original output, and λ is used as the weight decay so that the weights would not hold extreme values. Consequently, the weights of the network are updated at every e iteration by the following:

$$w_{j,e} \leftarrow w_{j,e-1} - \eta \frac{\partial L(y, y^*, w)}{\partial w_j}, \quad (8)$$

where η is the network's learning rate and is used to control the rate of changes to weights. Note that the updated network weights are hard to be interpretable just by their raw values. However, we can provide interpretability by observing the effect of the features on the target. This can be done by capturing the estimated target based on the marginal changes on one or more features as other features are held fixed. Upon construction of the nonlinear model, we can estimate the parameters of such model through the estimation of polynomial coefficients. We advise the use of such estimation to further reduce the computation time. Consequently, such nonlinear relation between target and features provides interpretability on the role of features which can then be leveraged in the forecast model.

As an example, we show the effect of each feature individually while keeping the other predictors fixed at their mean values. As seen in Figure 3, the effect of features is reported by keeping the other features at their mean value while slightly changing the predictor of the nonlinear model. What stands out in Figure 3 is the nonlinear model's ability to derive the nonlinear relation of multiple features to the target (e.g., hotel revenue). Moreover, one of the key nonlinear relations is the state's closure and opening policies over time for individuals and businesses. Such nonlinear relation of policy holds great importance in multi-step forecasts where the DemandNet needs to apply the policies in a dynamic manner for the task of multi-step prediction. To this end, DemandNet is able to store such nonlinear relations and leverage them for future predictions.

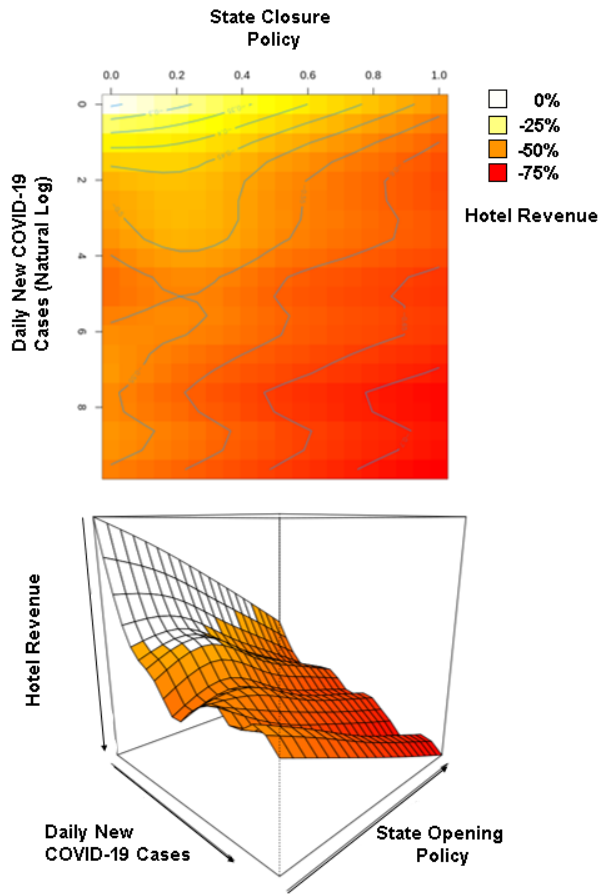


Figure 3. Nonlinear model of hotel revenue by the natural log of daily new cases and closure policy for the state of Florida. The closure policy changes from open (0) to the closure of all counties in a state (1).

3.3. Prediction Model

The two main outcomes of the previous components are the top correlated features of data and the nonlinear relation of policies with respect to the prediction targets. Thus, such information can be leveraged for the task of multi-step time series prediction. Particularly, DemandNet feeds the selected features into its prediction model and uses the state’s nonlinear relation with the prediction target in its control cell, namely, DemandCell. As shown in Figure 4, DemandCell is employed on top of well-known recurrent cells like LSTM. Specifically, it is designed to control the effect of state anti-contagion policies for the task of multi-step prediction, where the impact of such policies could have a long-lasting effect on consumer spending. The detailed procedure for the multi-step time series prediction is described below:

Given the input \hat{x} , the multi-step time series

prediction model is denoted as $y = \mathcal{F}_w(x)$, where x , y , and w stand for input data, output prediction, and model weights. As shown in Figure 4, the model \mathcal{F}_w can consist of any RNNs that deals with vanishing gradient problem such as LSTMs [13] or Gated Recurrent Networks (GRUs) [14]. GRUs are similar to LSTMs in a manner that the forget and input gates of LSTM get combined into a single “update” of GRUs gate. Additionally, the cell and hidden state are combined into the “reset” gate, which makes GRUs more efficient than LSTMs in terms of complexity.

We then employ DemandCell on top of the mentioned RNNs to control the dynamic anti-contagion policies over time. As shown in 4, there is a skip connection between the input and DemandCell, where the information of the state’s policy is used to control the output of the network. At the time step t where the prediction is required, the framework uses a skip connection to feed DemandCell with $P_{t:t+H}$, which includes a vector of variables containing information about policies for H days ahead. Note that DemandCell leverages the nonlinear model derived from the previous section to estimate the output accordingly based on the newly fed policies. Among the anti-contagion policies, we leverage the derived nonlinear relation of closure or reopening policy. Note that for predicting an unseen time series with unknown policies, such policies can be applied as a dummy variable to leverage the previously derived effects from every other state.

Next, it is important to equip the model to report the level of uncertainty so that the performance can be evaluated with trust for multi-step prediction. To this end, DemandNet uses the Monte Carlo Dropout mechanism to provide uncertainty in each of its predictions. The Monte Carlo Dropout is implemented after every recurrent (LSTM or GRU) layer, i.e., randomly dropping out each hidden unit with a certain probability of p . The probability of dropout varies from zero to one and determines the likelihood of connection from the hidden layer between RNN layers. The probability of this dropout is dynamically optimized for each prediction, and as the dropout is applied at random, each predicted value would differ from each other. Specifically, the prediction of the original target data y^* is done with the previously trained $\hat{\mathcal{F}}_w$, it generates K outputs y to obtain the prediction distribution $\{y_{(1)}, \dots, y_{(K)}\}$. By taking the mean of the derived distribution, we have:

$$\bar{y} = \frac{1}{K} \sum_{k=1}^K y_{(k)} \quad (9)$$

Then, the variance of the prediction distribution

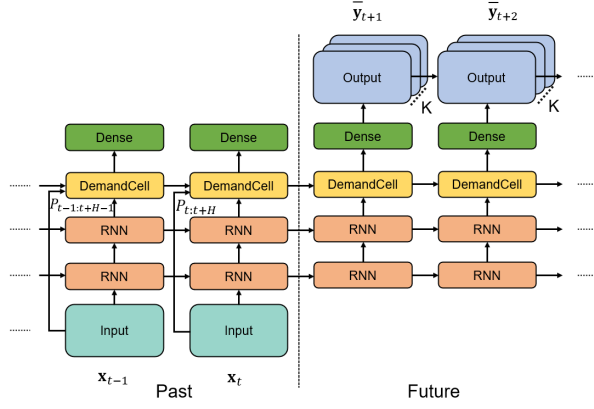


Figure 4. Multi-step time series prediction model. DemandCell is developed on top of existing RNNs (LSTMs or GRUs) to control the vector of dynamic state policies P_t and provide confidence in its final prediction (\bar{y}_t).

quantifies the prediction confidence:

$$\text{Var}(\hat{y}) = \frac{1}{K} \sum_{k=1}^K (y^{(k)} - y^*)^2 \quad (10)$$

To that end, we can use the mean of the distribution as a point prediction and its variance to report such prediction's confidence.

4. Experiments

This section reports the conducted experiments on DemandNet. Specifically, we provide a summary of the datasets in Section 4.1. Then, the baseline methods and evaluation metrics are reported in Sections 4.2 and 4.3 respectively. We then specify experimental settings in Section 4.4. Finally, we report the results of the experiments and discuss their interpretation in Section 4.5.

4.1. Datasets

In this experiment, we are selecting four real-world time series datasets. Our main dataset contains the hotel demand and revenue of 8 major tourist destinations in the US (e.g., Orlando, Los Angeles, ...). The dataset is provided by [15] and contains daily occupancy, demand, and revenue of the upper-middle class hotels.

The second dataset contains the daily report of new COVID-19 cases, deaths gathered by John Hopkins University [16].

The third dataset contains the mobility dataset

gathered by Google [17], which provides the relative changes of the resident's mobility prior to the pandemic (January 2020) in high-risk environments such as public transport and workplaces.

To enrich the datasets, we gathered dynamic exogenous variables such as the state's closure/open policy. We also gathered various static features such as the number of hospitals, GPD, and population, which were previously shown in Figure 2.

4.2. Methods for Comparison

The baseline methods for comparison include:

- Exponential Smoothing [18]: A traditional time series prediction method that uses an exponential window function to provide a smooth prediction.
- ARIMA [6]: A classical autoregressive integrated moving average method for time series prediction and is often used as a baseline.
- AE-LSTM [19]: An LSTM network that uses autoencoder for deep feature extraction and provides a deterministic prediction.
- Extreme-Event Forecasting [20]: A time series prediction model that uses Monte Carlo Dropout to extract the deep features of time series and provides uncertainty prediction.
- MQ-Transformer [21]: A transformer-based deep learning model that benefits from static features and future predefined labels. It also provides a state-of-art prediction in multiple datasets.

4.3. Metrics

For providing a comprehensive evaluation, we adopt three different evaluation metrics. The first two evaluation metrics are mean absolute error (MAE) and root mean square error (RMSE) that are scale-dependant measures. MAE is denoted as $\mathbf{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i^*|$, and RMSE is defined as $\mathbf{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - y_i^*)^2}$ where y_i is the predicted value and y_i^* is the original value. The third evaluation metric is the standard deviation (SD) that is correlated to the confidence of prediction and is denoted as $\mathbf{SD} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2}$ where \bar{y} is the mean of the predicted distribution.

Table 1. Description of Datasets Used in Experiments.

Dataset	Observations	Min	Max	Mean	SD
Hotel Revenue	25803 (Daily)	88.9K \$	4.34M \$	1.86M \$	691K \$
Hotel Demand	25803 (Daily)	1174	26927	5156	18799
Hotel Occupancy	25803 (Daily)	7.57%	99.09%	18.44%	74.01%

Table 2. Hyperparameters of DemandNet.

Parameter	Values
Batch Size	128
Learning Rate	1×10^{-5}
Weight Decay	1×10^{-6}
Multilayer NN Layers	2
RNN Hidden Units	128
RNN layers	2
Epochs	100
Training Time	10m

4.4. Experimental Settings

The models are implemented using Python 3.7 and tested on a cloud workstation that contains four Intel Xeon 2.3GHz CPUs, 32GB RAM, and one Nvidia P100 GPU. We conducted a grid search over all tunable hyperparameters on the held-out validation set for baseline methods and our proposed method and reported the results (Table 2).

From the baseline methods, we applied the widely used Exponential Smoothing and ARIMA by using Statsmodels library [24]. The nonlinear model and the prediction model are implemented by the TensorFlow [25] library. The hyperparameters of all models are tuned properly to provide the least amount of error in their prediction accuracy. Specifically, we conducted a grid search over all tunable hyperparameters on a 10% held-out validation set for the baseline methods and our framework. For both variations of DemandNet, the input window’s search ranges are chosen from {4, 8, 16, 32}. Note that enlarging the time window for more than 32-time steps will result in a marginal improvement in accuracy but substantially increase the training time for a dataset with a high dimension of features. The dimensions of the multilayer NN and RNN hidden layers are chosen from {32, 64, 128, 256}, and the number of layers is chosen from {1, 2, 4, 8}. Note that enlarging such numbers would increase the computation time.

Among NN models, the batch size and training epoch are set to be 128 and 200, respectively. We used the same set of essential features extracted

from the proposed feature selection mechanism for all baseline methods except for the ARIMA and Exponential Smoothing models, where they are not able to benefit from multidimensional features. Both variations of DemandNet can change the dropout probability dynamically for each prediction, and the SD is calculated based on $K = 100$ predictions.

In this study, we propose two sets of experiments. The first experiment is a standard 80 – 20 multi-step time series prediction evaluation method and is designed to evaluate the performance of DemandNet and baseline models. To this end, our dataset is divided into three subsets: training set (80%), validation set (10%), and the testing set (10%). The second experiment is designed to test the performance of DemandNet exclusively on completely untrained time series data. This experiment is applicable to evaluate the framework’s performance when a newly added set of time series data contains unknown features, or we need to abstain from the computational cost of training.

For both experiments, the performances of all models are not limited to one state or category, but the average observed losses for all test states and all six prediction categories. In this manner, our report is sufficient to justify the model’s performance for the total US consumers.

All architecture hyperparameters are set in a fairground. We used the same input features for the baseline methods except for the ARIMA model, which does not benefit from the multidimensional features.

4.5. Experimental Results and Discussions

To demonstrate DemandNet performance’s consistency, we provide two comprehensive evaluations: the aforementioned 80 – 20 testing (i.e., 80% trained) and the testing on a set of entirely untrained data. In both cases, we calculated the MAE, RMSE, and SD and discussed their interpretation.

Table 3 showcases the performance of DemandNet to compare with baseline in 80-20 testing phases. Compare to the baseline model, DemandNet-GRU achieved the highest accuracy for all horizon windows. However, the difference in its accuracy increases when the horizon forecast window increases.

Note that ARIMA and Exponential Smoothing

Table 3. Performance comparison of time series prediction methods.

Methods	Metrics	Forecast Horizon (Days)			
		10	20	40	80
Exponential Smoothing [22]	MAE	0.09423	0.12412	0.23631	0.53242
	RMSE	0.12324	0.15634	0.25341	0.55634
ARIMA [23]	MAE	0.05853	0.11487	0.23873	0.45387
	RMSE	0.07103	0.13268	0.25585	0.48695
EE-Forecasting [20]	MAE	0.04548	0.04695	0.05987	0.06122
	RMSE	0.04878	0.04921	0.06548	0.06587
	SD	0.00439	0.00527	0.00648	0.00631
MQ-Transformer [21]	MAE	0.02210	0.02261	0.03352	0.03731
	RMSE	0.02831	0.03724	0.04112	0.04312
	SD	0.00450	0.00551	0.00553	0.00605
DemandNet-LSTM (Ours)	MAE	0.01257	0.01655	0.01852	0.02263
	RMSE	0.01442	0.01831	0.02178	0.02434
	SD	0.00356	0.00351	0.00461	0.00612
DemandNet-GRU (Ours)	MAE	0.01302	0.01357	0.01424	0.02048
	RMSE	0.01504	0.01512	0.01734	0.02343
	SD	0.00398	0.00348	0.00451	0.00568

models are designed to provide deterministic results [10]. In contrast, EE-Forecasting and MQ-Transformer can provide the multi-step forecast with a level of confidence for each day. Among the baseline methods, the Exponential Smoothing and the ARIMA model’s error increase significantly as the forecast horizon increases. This is because both models are not able to leverage the essential features and adapt properly to sudden changes during the pandemic. Moreover, both methods cannot employ the essential features provided by DemandNet because they can only employ the target time series. Among NN-based models, only DemandNet and EE-Forecasting [20] benefit from the Monte Carlo Dropout mechanism. However, the EE-Forecasting is designed to use a static value of dropout probability through each experiment, while DemandNet adjusts this value dynamically for every forecast at each time step. Considering the selected features are the same for all these methods, we believe the lower SD is due to the dynamic dropout probability mechanism in DemandNet’s prediction model. Overall, the two proposed models, DemandNet-LSTM and DemandNet-GRU, consistently outperform the baseline NN models [21, 20], especially under longer horizon windows (40 and 80 days) due to their ability to incorporate state closure/open policy in their prediction. Considering all three evaluation metrics, DemandNet-GRU is best suited for the US consumer spending dataset due to reported higher confidence and accuracy.

The second experiment is designed to evaluate the performance of DemandNet on unseen and untrained

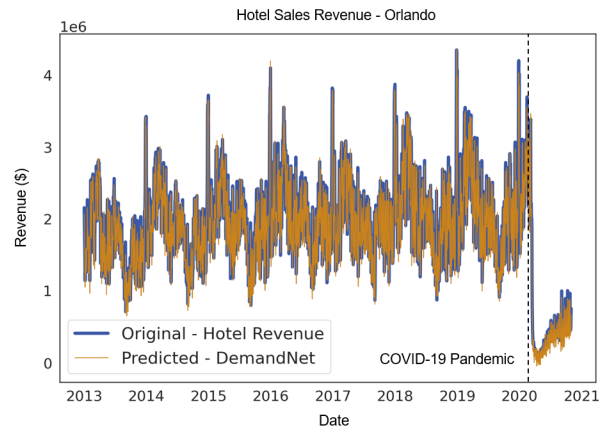


Figure 5. A sample forecast of DemandNet. Hotel sales of Orlando is the target. DemandCell aids in prediction of sudden drop during the COVID-19 pandemic.

data exclusively. Testing on untrained data, as shown in Table 4, holds importance when a new dataset is introduced to the network and might often hold unknown features. To this end, we kept away the complete data of four randomly chosen states in all three categories from the networks during the training phase and tested the performance of DemandNet on them. We believe that compared to 80-20 testing, such a testing procedure provides more significant insights into the model’s robustness.

Compared to 80-20 testing in Table 3, the reported error for unseen data is higher and has less confidence in the forecasts for all forecast horizons. This is due to

the model’s inability to train on the kept away time series and not being able to follow the trend properly. Between the two variations of DemandNet, DemandNet-LSTM provides higher accuracy at smaller horizons (10 days) but falls behind DemandNet-GRU in longer horizons. However, we believe that overall, DemandNet-GRU is a more robust model as it handles longer horizons with greater accuracy and less uncertainty.

Table 4. Performance of DemandNet-LSTM and DemandNet-GRU on completely unseen data.

Method	Metrics	Forecast Day Horizon			
		10	20	40	80
LSTM	MAE	0.03464	0.03645	0.04341	0.05653
	RMSE	0.03754	0.03883	0.04524	0.05836
	SD	0.00531	0.00643	0.00736	0.00833
GRU	MAE	0.03645	0.03345	0.04234	0.05533
	RMSE	0.03853	0.03953	0.04475	0.05765
	SD	0.00544	0.00653	0.00733	0.00856

Figure 5 demonstrates the prediction error of a typical case of a completely untrained time series dataset from one state (Florida). Both DemandNet-LSTM and DemandNet-GRU predictions follow the observed time series with great accuracy in the first days of the forecast horizon. However, the predictions’ errors increase as the horizon windows increase comparably to the early days of the predictions. As shown in Figure 5, the majority of loss occurs during the far horizon and is due to the framework’s inability to predict the trend accurately. Compared to the 80-20 experiments, using the second procedure (unseen testing) can be an acceptable tradeoff when a newly added time series requires punctual prediction or exogenous features are unavailable.

5. Related Works

There have been a surge in the development of general neural network (NN) frameworks for investigating the impact of COVID-19 on time series business data [26, 27, 28, 29]. The majority of these studies leverage NN prediction models to handle the large-scale nature of pandemic data. To derive the nonlinear properties of such data, various nonlinear functions need to be fitted [5]. Such a process requires a great amount of effort and is typically avoided for large-scale time series data with dynamic properties.

The authors in [7] proposed a transformer framework that achieves a state-of-art model for multi-horizon prediction for various complex time series datasets. Their method leverages static covariates and known inputs to improve its performance. Furthermore, they employ a gating mechanism that allows their framework to handle large-scale datasets. Additionally, they

leverage a temporal processing procedure where it enables the network to capture the long-and short-term relationships of both observed and known inputs (e.g., days of a week). Moreover, their model employs a seq-seq layer for short-term and attention mechanisms for long-term dependencies. However, our work is different from [7] due to the feature selection mechanism to derive highly correlated static features so that the network is able to provide greater accuracy and confidence in the predictions. Another practical advantage of DemandNet is that it provides a more interpretable and simpler nonlinear prediction model, which is far less computationally expensive than the transformers.

6. Conclusion

We propose a novel deep learning framework, DemandNet, to provide a robust multi-step time series forecast amid the COVID-19 pandemic. A feature selection mechanism is proposed to estimate the correlation between the COVID-19 pandemic and time series input data. This deep learning framework provides valuable insight into the social, health, and economic static features that affect the market demand, business operations, and COVID-19 cases. A multilayer neural network is then designed to derive the nonlinear relation of the key features to the observed input. Our results from the nonlinear model show that the negative impact of business closure policy on hotel demand and revenue to be higher than other location-specific exogenous variables (e.g., google mobility and COVID-19 daily new cases). Consequently, a novel RNN is developed to leverage the selected features and nonlinear model to provide a robust prediction with a high level of confidence. We evaluate the performance of DemandNet based on hotel sales revenue, demand, and occupancy. Compared with the state-of-the-art baseline methods, DemandNet provides superior prediction accuracy with a higher level of confidence. Our model can be expanded to model other business time series data under the influence of external shocks. Regarding future work, we plan to apply the DemandNet framework to understand the effect of various intervention policies on the COVID-19 transmissions in worldwide cities.

Acknowledgment

This work is supported by the US National Science Foundation under grants No. 1937833 and 2028481.

References

- [1] A. Brodeur, D. M. Gray, A. Islam, and S. Bhuiyan, "A literature review of the economics of covid-19," 2020.
- [2] S. Hsiang, D. Allen, S. Annan-Phan, K. Bell, I. Bolliger, T. Chong, H. Druckenmiller, L. Y. Huang, A. Hultgren, E. Krasovich, *et al.*, "The effect of large-scale anti-contagion policies on the covid-19 pandemic," *Nature*, vol. 584, no. 7820, pp. 262–267, 2020.
- [3] S. R. Baker, R. A. Farrokhnia, S. Meyer, M. Pagel, and C. Yannelis, "How does household spending respond to an epidemic? consumption during the 2020 covid-19 pandemic," tech. rep., National Bureau of Economic Research, 2020.
- [4] A. Kurmann, E. Lale, and L. Ta, "The impact of covid-19 on us employment and hours: Real-time estimates with homebase data," *May*. <http://www.andreukurmann.com/hb.covid>, 2020.
- [5] M. Långkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, pp. 11–24, 2014.
- [6] G. E. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *Journal of the American statistical Association*, vol. 65, no. 332, pp. 1509–1526, 1970.
- [7] B. Lim, S. O. Arik, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *arXiv preprint arXiv:1912.09363*, 2019.
- [8] S. Makridakis, R. J. Hyndman, and F. Petropoulos, "Forecasting in social settings: The state of the art," *International Journal of Forecasting*, vol. 36, no. 1, pp. 15–28, 2020.
- [9] F. Petropoulos and S. Makridakis, "Forecasting the novel coronavirus covid-19," *PloS one*, vol. 15, no. 3, p. e0231236, 2020.
- [10] I. Chung, S. Kim, J. Lee, K. J. Kim, S. J. Hwang, and E. Yang, "Deep mixed effect model using gaussian processes: A personalized and reliable prediction for healthcare," *arXiv preprint arXiv:1806.01551*, 2018.
- [11] D. Zimbra, A. Abbasi, D. Zeng, and H. Chen, "The state-of-the-art in twitter sentiment analysis: A review and benchmark evaluation," *ACM Transactions on Management Information Systems (TMIS)*, vol. 9, no. 2, pp. 1–29, 2018.
- [12] H. Akoglu, "User's guide to correlation coefficients," *Turkish journal of emergency medicine*, vol. 18, no. 3, pp. 91–93, 2018.
- [13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [14] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.
- [15] "Data-driven solutions empowering the hospitality industry."
- [16] E. Dong, H. Du, and L. Gardner, "An interactive web-based dashboard to track covid-19 in real time," *The Lancet infectious diseases*, vol. 20, no. 5, pp. 533–534, 2020.
- [17] A. Aktay, S. Bavadekar, G. Cossoul, J. Davis, D. Desfontaines, A. Fabrikant, E. Gabrilovich, K. Gadepalli, B. Gipson, M. Guevara, *et al.*, "Google covid-19 community mobility reports: Anonymization process description (version 1.0)," *arXiv preprint arXiv:2004.04145*, 2020.
- [18] C. Bergmeir, R. J. Hyndman, and J. M. Benítez, "Bagging exponential smoothing methods using stl decomposition and box-cox transformation," *International journal of forecasting*, vol. 32, no. 2, pp. 303–312, 2016.
- [19] A. Sagheer and M. Kotb, "Unsupervised pre-training of a deep lstm-based stacked autoencoder for multivariate time series forecasting problems," *Scientific Reports*, vol. 9, no. 1, pp. 1–16, 2019.
- [20] N. Laptev, J. Yosinski, L. E. Li, and S. Smyl, "Time-series extreme event forecasting with neural networks at uber," in *International Conference on Machine Learning*, vol. 34, pp. 1–5, 2017.
- [21] C. Eisenach, Y. Patel, and D. Madeka, "Mqtransformer: Multi-horizon forecasts with context dependent and feedback-aware attention," *arXiv preprint arXiv:2009.14799*, 2020.
- [22] A. M. De Livera, R. J. Hyndman, and R. D. Snyder, "Forecasting time series with complex seasonal patterns using exponential smoothing," *Journal of the American statistical association*, vol. 106, no. 496, pp. 1513–1527, 2011.
- [23] R. J. Hyndman, H. Booth, and F. Yasmeeen, "Coherent mortality forecasting: the product-ratio method with functional time series models," *Demography*, vol. 50, no. 1, pp. 261–283, 2013.
- [24] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python," in *Proceedings of the 9th Python in Science Conference*, vol. 57, p. 61, Austin, TX, 2010.
- [25] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pp. 265–283, 2016.
- [26] M. Fathi, M. Nemati, S. M. Mohammadi, and R. Abbasi-Kesbi, "A machine learning approach based on svm for classification of liver diseases," *Biomedical Engineering: Applications, Basis and Communications*, vol. 32, no. 03, p. 2050018, 2020.
- [27] M. Nemati, J. Ansary, and N. Nemati, "Machine-learning approaches in covid-19 survival analysis and discharge-time likelihood prediction using clinical data," *Patterns*, vol. 1, no. 5, p. 100074, 2020.
- [28] S. Lalmuanawma, J. Hussain, and L. Chhakchhuak, "Applications of machine learning and artificial intelligence for covid-19 (sars-cov-2) pandemic: A review," *Chaos, Solitons & Fractals*, p. 110059, 2020.
- [29] M. Nemati, J. Ansary, and N. Nemati, "Covid-19 machine learning based survival analysis and discharge time likelihood prediction using clinical data," *Available at SSRN 3584518*, 2020.