OmniaScience

Journal of Industrial Engineering and Management
JIEM, 2025 – 18(1): 193-213 – Online ISSN: 2013-0953 – Print ISSN: 2013-8423
https://doi.org/10.3926/jiem.6571

Bayesian-Optimized Ensemble Deep Learning Models for Demand Forecasting in the Volatile Situations: A Case Study of grocery Demand during Covid-19 Outbreaks

Nader Al-Theeb[1*], Hazem Smadi[2], Naser Al-Qaydeh[2]

[1]*Jordan University of Science and Technology (Jordan)*

[2]*Department of Industrial Engineering, Jordan University of Science and Technology (Jordan)*

[*]*Corresponding author: naaltheeb@just.edu.jo*
hjsmadi@just.edu.jo, nralqaydeh19@eng.just.edu.jo

Received: September 2023
Accepted: February 2025

Abstract:

Purpose: Lockdown and movement restrictions that imposed by governments have significantly changed customers' behavior, making the planning and decision-making processes more challenging. Providing accurate estimation of demand enables managers to take more successful decisions and allow optimizing inventory and resources; this is the main purpose of this study.

Design/methodology/approach: An ensemble model that is based on combining Bayesian-optimized Long Short-Term Memory (BO-LSTM) and Gated Recurrent Unit (BO-GRU). Experiments were carried out on actual dataset obtained from a company specialized in food industries during the volatile situation of Covid-19.

Findings: The proposed model significantly outperformed all hand-tuned ones and reduced the mean Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) by 2.80 % and 4.74 % compared to BO-LSTM and 3.14 % and 3.60 % compared to BO-GRU respectively. Furthermore, using BO algorithm for hyperparameters tuning improved the accuracy of forecasting.

Originality/value: The suggested model was statistically compared to its members in addition to other machine learning models using the t-test. Findings demonstrated the superiority of the proposed method over benchmark models.

Keywords: demand prediction, machine learning, ensemble model, bayesian optimization, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU)

To cite this article:

Al-Theeb, N., Smadi, H., & Al-Qaydeh, N. (2025). Bayesian-optimized ensemble deep learning models for demand forecasting in the volatile situations: A case study of grocery demand during Covid-19 outbreaks. *Journal of Industrial Engineering and Management*, 18(1), 193-213. https://doi.org/10.3926/jiem.6571

1. Introduction

Demand forecasting is both science and art that decision makers rely on to support the growth of the company, it allows efficient resource allocation, more effective management planning process, and help optimize the stock level. However, companies have struggled to provide accurate forecasts of demand due to increased volatility of consumer behavior, in addition to many factors such as weather changes and unexpected events like the Covid-19 pandemic (Jin, Zheng, Kong, Wang, Bai, Su et al., 2021).

In December 2019 a new infectious disease called Coronavirus 2019 (Covid-19) emerged in Wuhan, China. The responsible virus was named by the International Committee on Taxonomy of Viruses (ICTV) as SARS-CoV-2, which has a high level of transmission compared to similar viruses from the same family (coronaviridae) such as SARS-CoV (Shereen, Khan, Kazmi, Bashir & Siddique, 2020; World Health Organization, 2021). This disease is responsible for infecting and killing millions of people around the world. Until the writing of this paper, the total number of confirmed cases has reached 704,753,890, while the death toll reached 7,010,681 (World Health Organization, 2021).

Among the measures that governments have put in place was the lockdown, which has emerged as an effective way to control the spread of the virus (Atalan, 2020). Notwithstanding, it had major economic consequences and perilously impacted many manufacturing and service industries worldwide such as tourism, aviation, restaurants and bars, travel, and transportation (Minondo, 2021; Dey & Loewenstein, 2020; Agrawal, Jamwal & Guptza, 2020; Raouf, Elsabbagh & Wiebelt, 2020).

Covid-19 pandemic has imposed significant challenges for supply chains (SC) globally. Members in the SC have experienced shortage of labor, where infected workers, restrictions imposed on the number of workers allowed to be present in the facility, and the inability to get to work were the primary causes (Aday & Aday, 2020). Further, restrictions imposed on movement to and within the country impacted all modes of transportation, leading to the cessation of many industries due to the shortage of raw materials (Chowdhury, Paul, Kaisar & Moktadir, 2021). Consequently, these factors have negatively affected the flow of information through the chain and led to the loss of relationships that have been built over time. (Chowdhury, Sarkar, Paul & Moktadir, 2020; Pichler & Farmer, 2021).

Covid-19 epidemic changed people's lives and motivated them to invent new ways of consumption. Furthermore, consumers became under conditions to develop alternative ways to get their needs (Sheth, 2020; Roggeveen & Sethuraman, 2020). During the pandemic period, Customers' demand for basic products such as groceries has greatly increased, exceeding the stores' stock and the supplier's ability to replenish them (Saarinen, Loikkanen, Tanskanen, Kaipia, Takkunen & Holsmtröm, 2020). On the other hand, demand for unnecessary products has decreased because of the reduction in income. Additionally, the fear feelings of Covid-19 were the most influential on demand, especially at the beginning of the pandemic, however, it decreased over time (Rogggeveen & Sethuraman, 2020, Saarinen et al., 2020; Hoda, Singh, Rao, Ural & Hodson, 2020).

Implications of Covid-19 pandemic on SC activities, made the planning and decision-making processes more challenging (Chowdhury et al., 2021; Hoda et al., 2020). Management needs more responsive strategies that assist the SC to handle the resulting imbalance between supply and demand. Companies should be able to change quickly based on requirements and adapt to the current situation to survive (Dolgui, Ivanov & Sokolov, 2018). Further, they must understand the new drivers of demand and model them to get accurate forecasts (Chowdhury et al., 2021). Demand forecasting plays a key role in effective SC management, since providing accurate forecasts can guide business toward more successful decisions. Moreover, better estimation of consumer demand can assist in optimizing the inventory avoiding out-of-stock and over-stock situations, hence effectively meeting customers' needs.

These developments led to increase the demand volatility, making it very difficult for enterprises to accurately predict future demand using traditional forecasting methods such as Linear Regression and Autoregressive integrated moving average (ARIMA). These methods rely on building linear mapping between inputs and outputs, which makes them unable to capture complex nonlinear patterns within a dataset. Additionally, the accuracy of these methods decreases by increasing the forecasting horizon (Aslam, Lee, Khang & Hong, 2021).

Recently, machine learning algorithms have been successfully applied in various applications of time series forecasting. Feedforward Neural Network (FFNN) which is a type of Artificial Neural Networks (ANN) does not assume the distribution of data and it can model complex relationships within the dataset. Nonetheless, it is not able to handle temporal dependencies within the time series. On the other hand, Recurrent Neural Networks (RNN) is a special class of ANN that takes time dependencies into account, and it found to be more suitable for sequence data modeling than the FFNN (Abbasimehr, Shabani & Yousefi, 2020). Simple RNNs suffer from the vanishing and exploding gradient problems making it unable to handle long-range time dependencies. Long Short-Term Memory (LSTM) which is an extension to the RNN, in addition to the Gated Recurrent Unit (GRU) which is a simpler version of the LSTM were proposed to overcome the drawbacks of the vanilla RNN (Parmezan, Souza & Batista, 2019).

In addition to the learnable parameters, ANN contains hyperparameters which should be fixed before starting the training process. Hyperparameters selection is usually performed by an expert, where many trials are conducted, and the best performing configuration is selected. This process is time-consuming since the model must be trained and evaluated for each set of hyperparameters. Moreover, experts are expensive to hire and may not be available. What makes this process more difficult is the dependency of some hyperparameters on each other such as the learning rate and the batch size (Kandel & Castelli, 2020; Jastrezebski, Kenton, Arpit, Ballas, Fischer, Bengio et al., 2018), where increasing the value of one of them may require increasing or decreasing the value of the other to reach the best possible results. This problem could be bypassed by automating the process using more efficient ways. Many optimization algorithms have been developed and proven to provide a performance similar to outperform hand-tuned models (Bergstra, Bardenet, Bengio & Kégl, 2011). For instance, Bayesian Optimization (BO) has been widely used for hyperparameters tuning. What distinguishes this method is its ability to efficiently optimize expensive black-box functions by approximating them using a surrogate model that is cheaper to evaluate.

Weak predictors usually perform a local search, where they cannot cover a large area of the solution space. Furthermore, a single model may not be enough to represent all the relationships within the sequential data. Grouping several models to compose an ensemble can improve the performance of unstable learners (Qiu, Zhang, Ren, Suganthan & Amaratunga, 2014). One of the simplest ways to aggregate the results of the contributed models is by finding the average. However, this approach is sensitive to outliers (extreme values) (Choi & Lee, 2018). Alternatively, predictions could be combined by assigning a weight for each one of them based on a learning algorithm.

In this work, Bayesian-optimized ensemble model was proposed to forecast drinking water bottle packs demand during the volatile situation of Covid-19 pandemic. The major contribution of this paper is discussed in the following:

1. Weighted combination of Bayesian optimized LSTM and GRU models is applied to multistep-ahead demand forecasting. The ensemble consists of deep LSTM and GRU models, both were optimized using Bayesian Optimization (BO) method. Further, contributors were aggregated by training a blender model that weighed sum their predictions.

2. A comparisons study of the proposed ensemble model with RF, GBRT, FFNN, RNN, LSTM, GRU, in addition to its contributors, has been carried out to show the effectiveness of the proposed method. Most machine learning algorithms are stochastic in nature because of making use of randomness during the learning process. To achieve reproducibility and to provide fare comparison results, this paper uses the t-test to statistically compare the proposed method to its members as individuals, in addition to other benchmark models. Moreover, the optimized LSTM and GRU models were compared to the manual-tuned ones to investigate the effectiveness of BO algorithm for hyperparameters tuning.

The rest of the paper is organized as follows: section 2 discusses related works done in the field of time series forecasting. Then machine learning models are discussed in section 3. Section 4 presents the formulation of the proposed model, the data, as well as the methods used for comparison and forecasting. Section 5 explains the experimental results. Finally, section 6 concludes the paper and outlines future work.

2. Literature Review

Demand forecasting is one of the most important tasks of process planning, which can provide managers with the necessary guidelines for making appropriate decisions. One commonly used forecasting technique is Autoregressive Integrated Moving Average (ARIMA) which has been applied for various areas of application such as stock price (Ariyo, Adewumi & Ayo, 2014), gold price (Guha & Bandyopadhyay, 2016), and property crime (Chen, Yuan & Shu, 2008), food products (Fattah, Ezzine, Aman, Moussami & Lachhab, 2018), and perishable dairy products (Da Veiga, Da Veiga, Catapan, Tortato & Da Silva, 2014). However, due to the complexity of the real-life time series data, it is hard to fully understand the dataset (Khashei & Bijari, 2011) and determine whether it was related to linear or nonlinear process (Zhang, 2003). Further, ARIMA models are not able to detect large variations and to capture sudden changes in the data (Abbasimehr et al., 2020; Guha & Bandyopadhyay, 2016), hence, it is not able to provide satisfactory performance on complex time series (Khashei & Bijari, 2011; Zhang, 2003; Pai & Lin, 2005; Kofinas, Mellios, Papageorgiou & Laspidou, 2014).

Computational intelligence methods do not require prior knowledge about data distributions, also they showed good performance, outperforming statistical models when applied to complex and highly nonlinear time series, especially ANN which was able to beat both conventional time series and linear regression methods (Aday & Aday, 2020; Spiliotis, Makridakis, Semenoglou & Assimakopoulos, 2020; Smolak, Kasieczka, Fialkiewicz, Rohm, Siła-Nowicka, Kopańczyk, 2020; Jain, Kumar-Varshney, Chandra-Joshi, 2001). Special type of ANN called Recurrent Neural Networks which are more suitable for time series forecasting, due to its ability to model temporal dependencies within the sequential data (Abbasimehr et al., 2020; Parmezan et al., 2019). RNN showed its superiority over the FFNN (Carbonneau, Laframboise & Vahidov, 2008). Nevertheless, vanilla RNN suffer from the vanishing and exploding gradient problems which limit their ability to take advantage of longer sequences. Different gated architectures were designed to solve these problems such as Long Sort-Term Memory (LSTM) and Gated Recurrent Unit (GRU) (Parmezan et al., 2019).

In the study (Abbasimehr et al., 2020), a multilayer LSTM model was proposed to predict the future sales of a furniture company. The suggested model was configured using the Grid Search method and compared with several statistical and computational intelligence methods including: ARIMA, exponential smoothing, SVM, KNN, ANN, RNN, and single layer LSTM. Results demonstrated the ability of computational intelligence methods in handling complex real-world time series. Additionally, the proposed method showed the best performance. In Sahoo, Jha, Singh & Kumar, 2019), an LTSM model was developed to predict hydrological time series. The suggested model was compared to simple RNN and Naïve Method, performance of each model was evaluated using four different metrics namely RMSE, Nash-Sutcliffe efficiency, correlation coefficient, and MAE. Results showed the superiority of the gated structure, specifically LSTM over the simple RNN and the Naïve method. The study (Micheal, Hasan, Al-Durra & Mishra, 2022) proposed a novel deep learning model optimized Bi-directional long short-term memory to forecast univariate and multivariate time series data by integrating stacked LSTM layers. The model is optimized by Bayesian optimization through tuning of hyperparameters. Standard global horizontal irradiance and observed plane of array irradiance with metrological real solar data used in forecasting. The performance of the proposed algorithm was evaluated through uncertain weather conditions. The proposed model offered the best R^2 value of 0.99 for univariate and multivariate model.

A study (Habtemariam, Kekeba, Martinez-Ballesteros & Martinez-Alvarez, 2023) presented a model that is robust and optimizes long-short term memory network for forecasting wind power generation in Ethiopia. The model finds the best hyperparameter combination in a reasonable computational time using Bayesian optimization. The model was evaluated using MAE, RMSE, and MAPE metrics. While in (Usmani, Memon, Danyaro & Qureshi, 2024) the researchers developed a novel Optimized Multi-level Multi-type Ensemble model for forecasting power consumption. The model utilized different algorithms for time series forecast including exponential smoothing, LSTM, GRU and MLP. Bayesian and Tabu search optimization were used to tune parameters. The proposed model was able to predict power usage with an error of 22 %.

Conventional machine learning approaches depend on the features extracted by manual feature engineering of the dataset. Conversely, deep learning models can automatically extract relevant and better representations of the data (Ansari, Bartos & Lee, 2020). LSTM and GRU models also showed their ability to overcome traditional machine learning methods. For instance, (Wen, Zhou & Yang, 2020) proposes a deep learning model to forecast the load demand of residential buildings. Hourly measured load data were used in the study. Various models including: ARIMA, MLR, SVM, ANN, RNN, LSTM, and GRU were compared using different evaluation metrics, namely: RMSE, MAE, MAPE, and Pearson correlation. Results revealed that the GRU model is superior to the rest by providing forecasts with higher accuracy. Another case in (Kantasa-Ard, Bekrar & Sallez, 2019) where the author compared KNN, SVR, ARIMA and different structures of ANN to predict sugar consumption in Thailand, monthly sugar consumption rate from January 2015 to September 2018 were used in the study. Findings showed that the LSTM model was the best performing model.

The study (Liang, Lin, Deng, Mo, Lu, Yang et al., 2024) proposed a product market prediction framework. Artificial intelligence machine learning was incorporated in the framework that was built based on extreme gradient enhancement. To train and predict the ordering data for sales of time-series data, Bayesian optimized limit gradient boosting intelligent prediction model was used. The framework showed higher accuracy and faster speed prediction capabilities compared to traditional prediction methods.

Combining machine learning models could improve the forecasting results by increasing the chance of capturing different patterns. Qiu et al., 2014 propose an ensemble of deep learning belief network (DBN). The suggested model was structured as follows: first, 20 DBN models were trained, each of which with a different number of epochs. Then an SVM was used to map the predictions of the 20 DBN to the actual target value. The performance of the proposed model was compared to SVR, FFNN, DBN, and an ensemble of 20 FFNN based on two performance metrics namely RMSE and MAE. Findings revealed the superiority of the proposed model when tested on 7 different datasets. Additionally, showing the power of assembling multiple learners. Kamal, Bae, Sunghyun and Yun (2020) proposed a Deep Ensemble Recurrent Network (DERN) to predict Baltic Dry Index (BDI). Models in the ensemble (RNN, GRU, LSTM) were trained independently and the outcome is the weighted sum for their prediction. Further, these weights were learned using a neural network. Daily BDI data were sampled on a weekly basis using average values and used in the study. DERN was compared to ARIMA, MLP, Deep RNN, GRU and LSTM using MAE, MAPE, and RMSE metrics. Results demonstrated that the proposed method outperformed single structured models on both short-term and long-term prediction. Many other ensembles were proposed in the literature which consisted of diverse types of forecasting models that aggregated differently. These ensembles showed promises improvements and achieved better results in comparison to single structured predictors (Jin, Ye, Ye, Wang, Cheng & Yan, 2020; Huang, Zhang, & Song, 2021; Akyuz, Uysal, Bulbul, & Uysal, 2017; Ishaq & Kwon, 2021; Xenochristou & Kapelan, 2020; Tan, Yuan, Li, Su, Li & He, 2019; Qiu, Ren, Suganthan & Amaratunga, 2017). The study (Ozaki, Ooka & Ikeda, 2021) discussed the relationship between a machine learning model's prediction accuracy and its hyperparameters. Grid search, random search, and Bayesian optimization tuning methods were used. All tuning methods reduced the RSME to less than 50 % compared to non-optimized tuning.

In summary, the previous review of the literature confirmed the superiority of deep learning methods, especially LSTM and GRU which outperformed statistical methods, traditional machine learning models, FFNN, and Simple RNN. Furthermore, studies showed that aggregating multiple models can improve the forecasting quality. Still, most studies of demand forecasting in the literature were carried out during the normal situation, where the demand was more stable and easier to anticipate. Furthermore, many of the proposed models were tuned by using trial and error methodology, which does not guarantee access to the best possible model's configuration. Moreover, these models were benchmarked by performing a single run comparison, which is an unfair method, since the model's convergence point depends on the initial random values, resulting in a different model at each run.

Accurate time series forecasting is crucial for decision making processes and reducing the future uncertainty. Hence, this work proposes a deep ensemble model which consists of LSTM and GRU models. The BO algorithm was used to configure both models that contribute to the ensemble. The proposed model was statistically compared to its members, in addition to other machine learning models using the t-test. This research elucidates the effectiveness of ensemble models, and the importance of taking it into consideration in complex and volatile situations. Further, this work will convey valuable information for future research that will explore various aggregation techniques.

3. Theoretical Background

In this section, a theoretical background about machine and deep learning models will be presented. Some of these models will be used in this research, and others will be used as benchmarks to compare with this work.

3.1. Ensemble Machine Learning Models

Decision Tree (DT) is a supervised machine learning algorithm that can be used for both classification and Regression problems (Tugay & Oguducu, 2020; Pedregosa, Varoguaux, Gramfort, Michel, Thirion, Grisel et al., 2011; Géron, 2019). DT needs a minimal amount of data preprocessing and can produce accurate results. However, DT is unstable and sensitive to small variations in the training dataset. Moreover, in many cases, a single tree may be not enough to describe the relationships within the dataset. These problems could be solved by aggregating multiple DT into an ensemble such as the Random Forest (RF) (Lahour & Slama, 2015), and Gradient boosting Regression Tree (GBRT) (Géron, 2019).

3.2. Deep Learning Models

3.2.1. feedforward neural network

Feedforward Neural Network is a class of ANN that imitates biological neural networks. FFNN consists of neurons (also called nodes) that are interconnected and organized in layers. Each node in a layer is connected to all neurons in the next layer, and these connections are associated with weights. FFNN Includes an input layer, one or more hidden layer, and an output layer. The input layer receives the input from the external environment, the output layer communicates the output to the environment, and hidden layers encode the relationships between the input and the output. Neural networks are learned by exposure to training examples and target values, where learning is achieved by adjusting the weights value in a direction that minimizes the error between model predictions and the true targets (Burkov, 2019; Chollet, 2021).

3.2.2. Simple RNN

Information flows through the FFNN in one direction, from the input layer, through the hidden layers, to the output layer without any feedback loops (Carbonneau et al., 2008). Furthermore, the inability of FFNN to handle historical data dependencies, makes it unsuitable for sequence data modeling (Bedi & Toshniwal, 2019). On the other hand, RNN, which is a special type of ANN accounts for temporal dependences within the dataset (Abbasimehr et al., 2020). RNN unit takes information from previous steps and utilizes it to predict the next step. However, simple RNN suffers from the vanishing gradients problem, where gradients have a hard time propagating and adjusting earlier weights, consequently, making the RNN stores previous information for a short period of time and is unable to make use of long-range dependencies (Bedi & Toshniwal, 2019; Siami-Namini, Tavakoli & Namin, 2018).

3.2.3. Long Short-Term Memory

Long Short-Term Memory is an extension of the traditional RNN proposed by Hochreiter and Schmidhuber in 1997 to overcome the weaknesses of simple RNN (Abbasimehr et al., 2020). LSTM hidden layer consists of LSTM units as shown in Figure 1, every unit works at a distinct time step and passes its output to the next unit until the last one which produces the output. The LSTM unit contains three controlling gates namely input gate, forget gate, and the output gate, which control the flow of information through the LSTM layer. Additionally, the unit contains a memory cell that can maintain information for a long period of time. Figure 2 shows the structure of an LSTM block.

For a time series *x*, the LSTM unit updates the input cell state c_t and output a hidden state h_t at each time step *t*, according to the following equations [55].

$$i_t = \sigma\ (w_{ih}h_{t-1} + w_{ix}x_t + b_i) \tag{1}$$

$$f_t = \sigma\ (w_{fh}h_{t-1} + w_{fx}x_t + b_f) \tag{2}$$

$$o_t = \sigma\ (w_{oh}h_{t-1} + w_{ox}x_t + b_o) \tag{3}$$

$$\tilde{c}_t = tanh\ (w_{ch}h_{t-1} + w_{cx}x_t + b_c) \tag{4}$$

$$c_t = i_t * \tilde{c}_t + f_i * c_{t-1} \tag{5}$$

$$h_t = o_t * tanh\ (c_t) \tag{6}$$

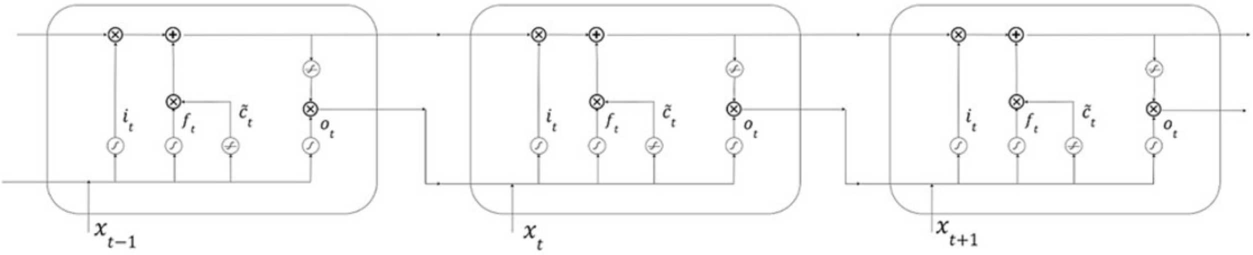

Figure 1. The structure of LSTM layer

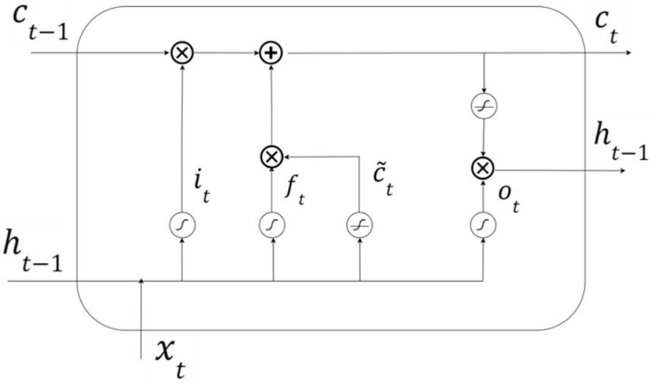

Figure 2. The structure of LSTM unit

In equations (1) to (6), i_t, f_t, o_t represent the input gate, forget gate, and the output gate respectively, σ denotes the logistic sigmoid function, w_{ih}, w_{ix}, w_{fh}, w_{fx}, w_{oh}, w_{ox}, w_{ch}, and w_{cx} stands for the weight matrices, *tanh* is the hyperbolic tangent activation function, b_i, b_f, b_o and b_c are the bias parameters, and $\tilde{c}_t$ is a candidate value to be added to the cell state c_t. LSTM has been commonly used for sequential data modeling due to its ability to handle long-term dependencies. The gating mechanism of the LSTM cell allows to maintain its state value over a long time by regulating the flow of information into and out of the cell and to be kept or discarded. The input gate will specify the relevant information to keep, the forget gate will determine the information that should be deleted from the previous time step, and the output gate will determine the information that will be used to generate the cell output.

3.2.4. Gated Recurrent Unit

Gated recurrent unit is a less complicated variant of the LSTM introduced by Cho, Van Merriënboer, Gulcehre, Bahdanau, Bougares, Schwenl et al. (2014). GRU aims to solve the vanishing gradient problem of the vanilla RNN. Unlike the LSTM, the GRU unit consists of only two gates to control the flow of the signal namely update gate and reset gate, making it more computationally efficient and cheaper to train. Moreover, the GRU unit does not contain a separate memory cell. The structure of GRU block shown in Figure 3.

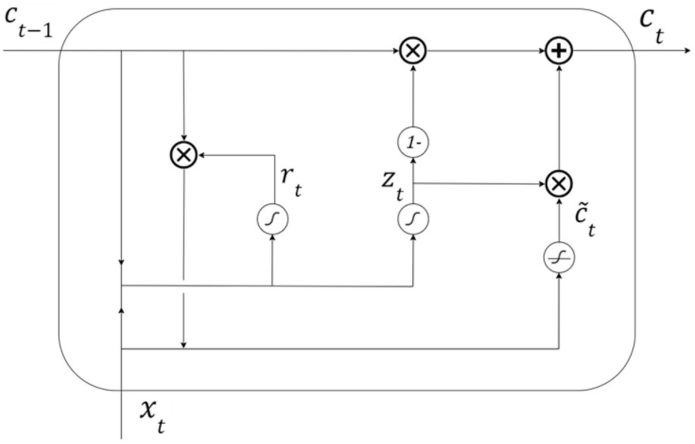

Figure 3. The structure of GRU unit

Mathematically (Zeroual, Harrou, Dairi & Sun, 2020), the forward learning of an GRU is as follows:

$$z_t = \sigma(w_{uc}c_{t-1} + w_{ux}x_t + b_u) \tag{7}$$

$$r_t = \sigma(w_{rc}c_{t-1} + w_{rx}x_t + b_r) \tag{8}$$

$$\tilde{c}_t = tanh\ (w_{cc}(r_t * c_{t-1}) + w_{cx}x_t + b_c) \tag{9}$$

$$c_t = z_t * \tilde{c}_t + (1 - z_t) * c_{t-1} \tag{10}$$

Where z_t and r_t are the output results for the update gate and reset gate respectively. σ is the logistic sigmoid function, w_{uc}, w_{ux}, w_{rc}, w_{rx}, w_{cc}, and w_{cx} represent the weight matrices, b_u, b_r, and b_c donate the bias parameters, c_{t-1} is the previous cell state, and $\tilde{c}_t$ is a candidate for replacing c_{t-1}. The next section explains which model is utilized in this study.

4. Methodology

In this study, five main phases will be used to achieve the purpose, as follows.

4.1. Data

In this work, actual daily sales of 1.5-liter drinking water bottle packs were used for modeling. The dataset obtained from a company specialized in food industries, headquartered in Amman, Jordan. It is one of the leading companies in the field of bottling and distributing mineral drinking water in Jordan, and it covers all parts of the country. The dataset consists of a univariate time series which was collected during Covid-19 pandemic period, covering 533 days from March 1st, 2019 to August 14th, 2020.

4.2. Data Preparation

Real world datasets may contain noise, duplicate, and missing values, resulting in a poor-quality model and unsatisfactory performance. Hence, data preparation is an important step, by which raw data is refined before being fed to the algorithm. Data preparation process includes several sub-steps that differ according to the type of data and the problem to be dealt with.

Four-step process was employed to make the data suitable for modeling. First, missing values were found and replaced using forward filling technique. Next, data were split into a training set which was utilized to develop the models, and a testing set that was used to assess the quality of models' predictions. As shown in Figure 4, the historical training set consists of the demand from March 1st, 2019, to July 31st, 2020, and the testing set extends from August 1st, 2020 to August 14th, 2020. Then, the training set was standardized according to a distribution of mean 0 and standard deviation of 1 to improve the learning process and ability of the model to smoothly digest the data.

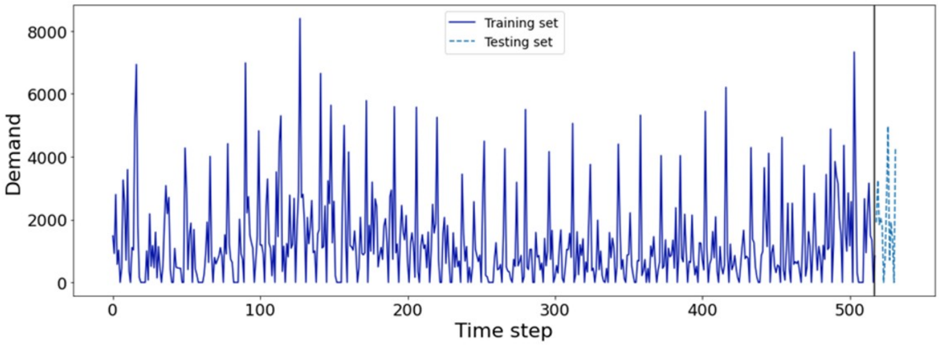

Figure 4. Training and testing splits of drinking water bottle packs demand dataset

Finally, sliding window technique was applied to transform the time series into an appropriate form, where a window of size 10 slide over the time series to extract features and labels. The window size was selected based on trial-and-error method where different sizes from 1 to 30 were tested. The feature is a window of consecutive values (order preserved) from the series, while the label is the next value. This process restructures the time series, making it suitable for supervised learning problems. Figures 5 demonstrate a graphical representation of the windowing process for both sets.

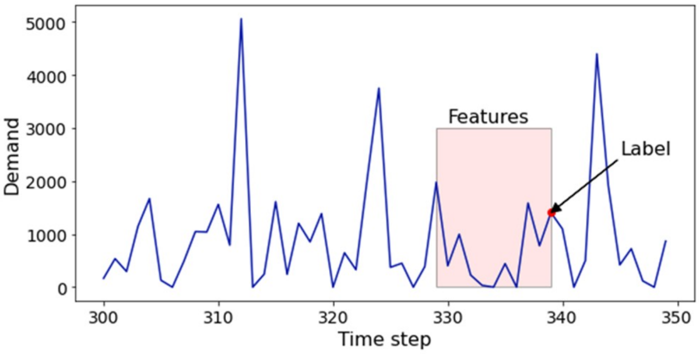

Figure 5. Sliding window applied to the drinking water bottle packs sales with a window size of 10

4.3. Proposed Method

In this section, the proposed model will be explained, and how the hyperparameters are optimized is discussed.

4.3.1. The Proposed Model

LSTM and GRU models were proposed to combat the problem of vanishing gradient found in simple RNN using the gating mechanism which allows them to learn long term dependencies. However, it is not possible to generalize which one is the best and most appropriate for a particular problem or a field of application, each one has its own way of information flow management using different internal operations and gating mechanism. The proposed model considers a combination of Bayesian-optimized LSTM (BO-LSTM) and GRU (BO-GRU) into an ensemble with the purpose of reducing the forecasting error.

In regression problems, one of the simplest ways to build an ensemble is to combine the predictions of different regressors using a summary statistic, such as the mean. While in this work, instead of using a trivial function, a blender model was trained to perform the aggregation. As shown in Figure 6, the suggested Blended-LSTM-GRU model consists of Bayesian-optimized LSTM (BO-LSTM) and GRU (BO-GRU) models, which were independently trained using the same pre-processed data. Next, trained models were used to generate predictions on the training data, which were utilized to train the blender parameters w_1 and w_2. The blender is the simplest form of neural network architecture, which consists of only one neuron. In the testing phase, predictions of BO-LSTM and BO-GRU represented by P_1 and P_2 were aggregated into one final prediction P by applying a weighted sum as shown in equation (11).

$$P=w_1\ P_1+\ w_2\ P_2 \tag{11}$$

Where w_1 is the weight of the BO-LSTM prediction and w_2 is the weight of the BO-GRU prediction. Traditional machine learning models (RF and GBDT) were implemented in python using Scikit-learn library (Pedregosa et al., 2011), while all deep learning models were developed using TensorFlow platform (Abadi, Agarwal, Bahram, Brevdo, Chen, Citro et al., 2016). Experiments have been conducted in an environment with AMD Ryzen 7 5800H with Radeon Graphics CPU, Nvidia RTX 3060 laptop GPU, 1 TB of SSD storage, and 16 GB of RAM.

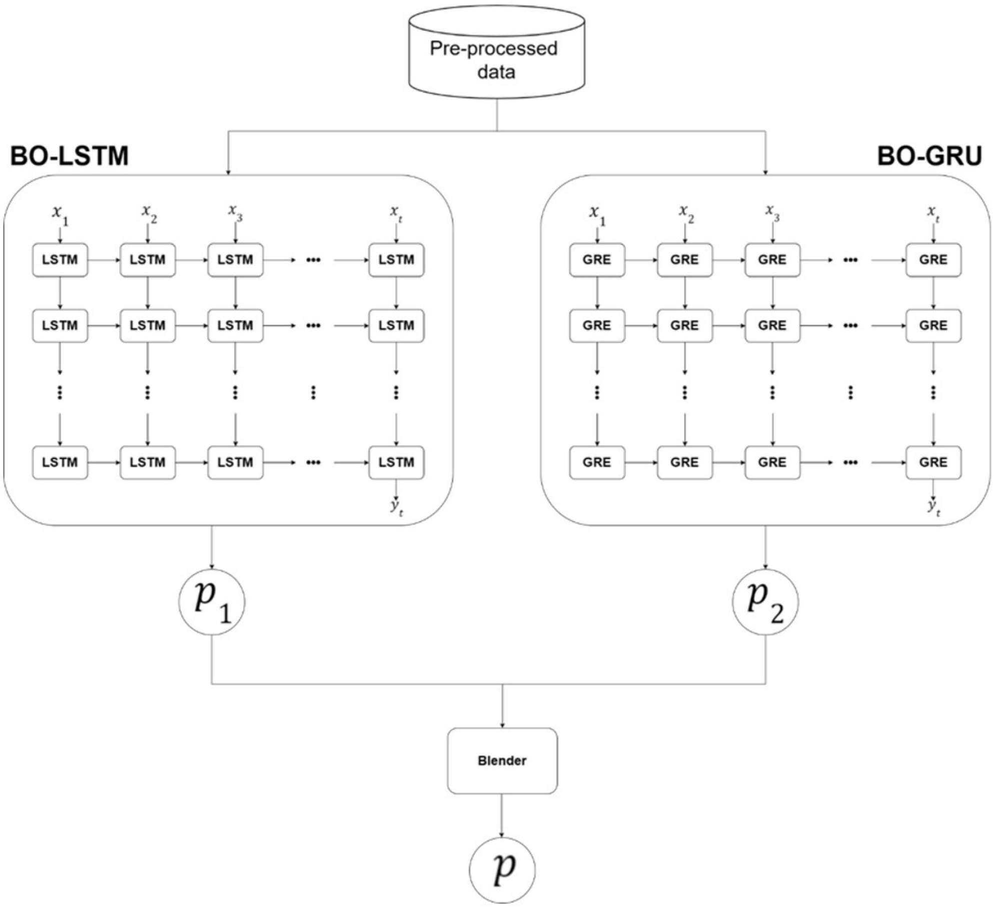

Figure 6. Blended-LSTM-GRU model structure

Where w_1 is the weight of the BO-LSTM prediction and w_2 is the weight of the BO-GRU prediction. Traditional machine learning models (RF and GBDT) were implemented in python using Scikit-learn library (Pedregosa et al., 2011), while all deep learning models were developed using TensorFlow platform (Abadi et al., 2016). Experiments have been conducted in an environment with AMD Ryzen 7 5800H with Radeon Graphics CPU, Nvidia RTX 3060 laptop GPU, 1 TB of SSD storage, and 16 GB of RAM.

4.3.2. Hyperparameters Optimization

Bayesian Optimization (Chowdhury et al., 2021) is a sequential design strategy used to optimize black-box expensive functions. In contrast to the manual tuning process where a new model must be built, trained, and evaluated each time a new set of hyperparameters is proposed, BO approximates the objective function using a surrogate model which is cheaper to evaluate. Various forms of the BO algorithm which differ in the way that they model the actual function (surrogate model building approach) and in the criterion that is optimized to get the next values for evaluation (the acquisition function).

The surrogate model is a probabilistic representation of the actual objective function. Since the actual distribution of the objective function scores is not known, a sample of (hyperparameters values, actual function score) pairs is generated and used to build and train the surrogate model. To select the next point to evaluate, acquisition function (also called selection function) is used. The next query point x^* is the one that maximizes that acquisition function. Next, x^* is evaluated on the actual function and the score is obtained. Since the surrogate model was trained in a history of (hyperparameters values, actual function score) pairs, by adding a new pair, the surrogate model could be updated. These steps are repeated until the provided number of iterations is reached. Figure 7 shows the general flowchart of the BO algorithm.

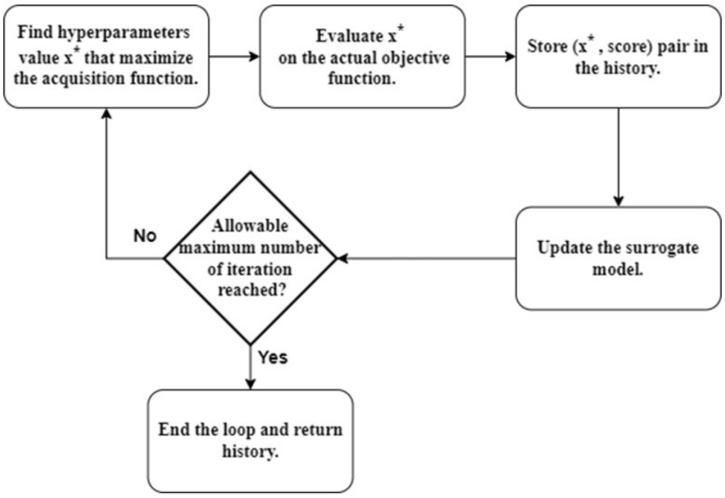
Find hyperparameters value x* that maximize the acquisition function.
Evaluate x* on the actual objective function.
Store (x*, score) pair in the history.
Update the surrogate model.
Allowable maximum number of iteration reached?
No
Yes
End the loop and return history.

Figure 7: Procedure of Bayesian Optimization algorithm

One of the common criteria is Expected Improvement (IE) which is defined as:

$$EI_{y^*}(x) = \int_{-\infty}^{\infty} max(y^* - y, 0)\, p(y|x)\, dy \tag{12}$$

Where: x is the proposed set of hyperparameters values, y represents the value of the actual function on x, and $p(y|x)$ is the surrogate probabilistic function. Two common approaches to build the surrogate model: Gaussian Process (GP) and Tree-structured Parzen Estimate (TPE). GP will model $p(y|x)$ directly by using a history of (hyperparameters values, actual function score) pairs to build multivariate Gaussian distribution, while TPE which will be used in this research, model $p(x|y)$ and $p(y)$. TPE define the $p(x|y)$ as:

$$p(x|y) = \begin{cases} l(x) & if\ y < y^* \\ g(x) & if\ y \geq y^* \end{cases} \tag{13}$$

TPE chooses y^* to be some quantile γ of the observed y values which separate the current observations into two clusters. Two density distributions are built: $l(x)$ when y is less than the threshold y^*, and $g(x)$ when y is greater or equal to the threshold y^*. In other words, TPE uses the observations that gives a loss lower than the threshold y^* to build *l*(*x*), and the rest are used to build *g*(*x*). No specific model for *p*(*y*). After some modifications and Using Bayes rule, EI becomes as follows:

$$EI_{y^*}(x) = \frac{\gamma y^* l(x) - l(x)\int_{-\infty}^{y^*} p(y)\,dy}{\gamma l(x) + (1-\gamma) g(x)} \propto \left(\gamma + \frac{g(x)}{l(x)}(1-\gamma)\right)^{-1} \tag{14}$$

As mentioned earlier, EI is used to select the next values to evaluate, these values are the ones that maximize it. In order to maximize EI, points *x* with high probability under *l*(*x*) and low probability under *g*(*x*) are selected. In other words, it is preferable to choose points that are more likely to be under *l*(*x*). Even though the BO algorithm spends time to give the next proposed values that maximize the EI, it's more efficient due to the fact that this method uses an informed manner to propose the next values to the actual expensive function.

GRU and LSTM networks were optimized using the BO method. These models contain many hyperparameters including but not limited to the learning rate, batch size, loss function, number of hidden layers, number of nodes in each hidden layer, the optimizer, and the number of epochs. Based on the test results on the manually tuned models, the optimizer was fixed on Adam, and the remaining hyperparameters were selected using the BO method. First, the ranges of hyperparameters values should be defined which represent the search space for the BO algorithms to explore. Table 1 provides the ranges of hyperparameters values that are used in the experiments.

The performance of each hyperparameters combination was evaluated using the RMSE and MAE. A python function that accepts the hyperparameters values and returns the RMSE and MAE scores for that combination was built. This function represents the objective function that was optimized using the BO algorithm. Experiments were carried out using an open-sourced python library for BO called Hyperopt (Bergstra, Yamins & Cox, 2013). Hyperopt has a simple user interface, and the search space could consist of continuous, ordinal, or categorical variables providing a greater flexibility for the optimization process. Additionally, BO was ran for 35 iterations on each model.

Hyperparameter	Range
Activation function	[Tanh, ReLU]
Loss function	[Huber, MSE, MAE]
Batch size	[2, 32], step = 2
Learning rate	[0.0001, 0.01]
Number of hidden layers	[1, 2, 3]
Number of neurons	[32, 64, 128, 256, 512]
Number of epochs	[100, 550], step = 10

Table 1. Range of hyperparameters for the BO method

4.4. Prediction Strategy

Recursive Multi-step Forecast strategy was used to extrapolate the time series multiple steps ahead. This method uses a single model, which has been trained to make one-step ahead prediction. To illustrate, after training the model, the last *k* observations are extracted from the training dataset, the value of *k* is equal to the window size used in the training phase. The extracted array of points *x* then used to predict one-step ahead in the future. To predict the next step, the first element in *x* was dropped and the predicted previous value was appended to *x*. Using the new modified *x*, the model will predict the next step. This process continued until the required forecasting horizon was reached. Simply put, the model makes use of predictions of prior time steps as an input to forecast the following time step.

4.5. Models Comparison

Different runs of the same model (same code) that trained on the same data set may produce different results (Beam, Manrai & Ghassemi, 2020). This behavior could be attributed to the stochastic nature of the algorithms, where various sources of variability exist. For instance, at each run, weights are initialized into different random values leading to different internal decisions in the model during the training process.

In this study, models' performance was compared statistically using the t-test. Each model was run 30 times, at each run, randomly selected seed value was used. The performance at each run was recorded to obtain samples of performance measures for each model, since the t-test assumes that the samples are approximately normally distributed. Normality was checked numerically using the D'Agostino and Pearson's normality test with a significance level of 0.05 (D'agostino & Pearson, 1973). If the p-value is less than 0.05, the null hypothesis (the sample came from a normal distribution) is rejected, while a p-value higher than 0.05 indicating that the sample came from a normal distribution. Box-Cox Transformation (Box & Cox, 1964), with Lambda equal to 0 was used to transform the data to make it fit a normal distribution.

The following metrics were used to evaluate the forecasting techniques: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE).

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \tag{15}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2 \tag{16}$$

Where y_t are the actual values, $\hat{y}_t$ are the estimated values, and n is the number of observations.

5. Results and Discussion

Covid-19 pandemic has disrupted the consumption pattern, making demand forecasting more challenging. Recently, the trend towards using deep learning models for time series forecasting has increased, since these models, especially recurrent neural networks, are assumption free, able to handle nonlinear complex problems, and take time dependencies into account. Many machine learning algorithms, which differ in the way they perceive and process the input data. Each algorithm may learn different features by making different assumptions about the prediction problem, leaving other patterns that couldn't detect undiscovered. Combining different machine learning models may improve the overall performance, outperforming the contributing members separately.

Hyperparameters	BO-LSTM	BO-GRU
Number of hidden layers	2	2
Number of neurons	(256, 256)	(256, 256)
Learning rate	0.00155	0.00122
Batch size	32	32
Activation function	Tanh	Tanh
Optimizer	Adam	Adam
Loss function	MAE	MAE
Number of epochs	490	250

Table 2. Optimal hyperparameters value for BO-LSTM and BO-GRU

This work proposed the Blended-LSTM-GRU model which consists of three main parts including the BO-LSTM, BO-GRU, and the blender. The same pre-processed dataset was used to independently train BO-LSTM and BO-GRU models, and both were automatically tuned using the BO algorithm. Table 2 shows the optimal hyperparameters value for both BO-LSTM and BO- GRU. The suggested model was statistically compared to 6 benchmark models namely RF, GBRT, FFNN, RNN, GRU, LSTM, in addition to its contributors.

Model selection usually involves evaluating many different machines learning algorithms and comparing them based on their performance on an unseen testing dataset. The model that achieves the best results is then selected as the final model. This approach can be misleading, and it can lead to choosing a suboptimal model. Summary statistics and statistical tests such as the t-test could give more accurate comparison than using a single model performance measure.

Standard deviation (std) and range can provide information about the variability of the dataset: showing how the data are spread around the mean, for instance, high std indicates more spread-out data, while low std means that the data is clustered around the mean. From Table 3, the high variability of deep learning models can be observed by looking at the std and range values. Each run of the same model (same code and hyperparameters value) provided different results, this behavior can be attributed to the random nature of these models as mentioned in section 2 and 4.5, resulting in an unfair comparison when using single-run method.

Samples of RMSE and MAE values for each method were obtained by running the model 30 times, each with a different random state. D'Agostino and Pearson's normality test with a significance level of 0.05 was used to determine if the samples are normally distributed. It was found that all p-values for RMSE samples were lower than 0.05 which means that the null hypothesis (sample is normally distributed) is rejected and the RMSE samples skewed and deviated from a normal distribution. Wherefore, they were transformed using Box-Cox Transformation with a Lambda equal to 0 (logarithmic transformation). All p-values for the MAE samples were higher than 0.05 indicating that the null hypothesis failed to reject, and the MAE samples were normally distributed. The t-test was used to compare the performance of the forecasting methods considered. Table 3 shows the mean, standard deviation, and range of RMSE and MAE samples for each model. Tables 4 and 5 illustrate the resulting p-values of the t-test.

Model	RMSE			MAE		
	Mean	Std Dev	Range	Mean	Std Dev	Range
GBDT	1370.39	72.84	183.16	1025.52	37.45	127.80
RF	1330.67	17.74	74.97	970.85	11.76	53.57
FFNN	1327.19	147.18	521.71	968.69	113.14	431.32
RNN	1395.44	137.16	580.28	1001.55	126.20	629.14
LSTM	1311.45	209.04	821.89	980.56	192.41	755.10
GRU	1477.61	208.53	852.34	1061.73	157.96	728.13
BO-LSTM	1234.78	227.50	778.70	905.59	184.85	702.21
BO-GRU	1259.79	129.32	539.91	909.88	130.52	580.76
Blended-LSTM-GRU	1200.04	191.03	739.75	877.15	133.20	517.85

Table 3. Mean, standard deviation, and range of RMSE and MAE samples

Forecasting technique	GBRT	RF	FFNN	RNN	GRU	LSTM	BO-GRU	BO-LSTM	Blended-LSTM-GRU
GBDT									
RF	0.0040								
FFNN	0.0585	0.3388							
RNN	0.7697	0.9909	0.9685						
GRU	0.9878	0.9995	0.9981	0.9424					
LSTM	0.0409	0.1823	0.3082	0.0229	0.0018				
BO- GRU	0.0000	0.0012	0.0377	0.0000	0.0000	0.1772			
BO- LSTM	0.0007	0.0047	0.0215	0.0004	0.0000	0.0792	0.2088		
Blended-LSTM-GRU	0.0000	0.0001	0.0024	0.0000	0.0000	0.0194	0.0582	0.2944	

Table 4. Probability that the two samples (RMSE) come from the same distribution

Forecasting technique	GBDT	RF	FFNN	RNN	GRU	LSTM	BO-GRU	BO-LSTM	Blended-LSTM-GRU
GBDT									
RF	0.0000								
FFNN	0.0064	0.4594							
RNN	0.1654	0.9014	0.8496						
GRU	0.8827	0.9985	0.9938	0.9428					
LSTM	0.1109	0.6064	0.6122	0.3125	0.0422				
BO-GRU	0.0000	0.0075	0.0359	0.0043	0.0000	0.0535			
BO-LSTM	0.0006	0.0314	0.0612	0.0123	0.0005	0.0678	0.4595		
Blended-LSTM-GRU	0.0000	0.0002	0.0033	0.0003	0.0000	0.0103	0.1743	0.2521	

Table 5. Probability that the two samples (MAE) come from the same distribution

There is no unified answer to how many layers are the most appropriate or how many neurons are the best for all datasets. Additionally, depending on the complexity, fewer number of layers may produce an underfitting results, while too many layers may overfit the learning dataset reducing the model ability to generalize to new unseen data. It was noticed that running the same algorithm with different hyperparameters value led to different results, since hyperparameters affect the model behavior and its ability to detect patterns. Moreover, as mentioned in section 1, model hyperparameters may depend on each other, which also makes the process of selecting them manually more challenging. From the results it can be seen that BO can be used to determine the right combination of hyperparameters that maximizes the performance of the model on the considered dataset in an efficient way.

Results of the experiment showed that the proposed Blended-LSTM-GRU was the most successful by providing the lowest mean RMSE and MAE. The Blended-LSTM-GRU overcame its components, where compared to the BO-LSTM, the mean RMSE and MAE were reduced by 2.80 % and 3.14 % respectively, while compared to the BO-GRU, the reduction in mean RMSE and MAE were 4.74 % and 3.60 % respectively.

BO-LSTM and BO-GRU were the second and third most accurate forecasting methods respectively. However, there was no statistically significant difference between them in terms of both RMSE and MAE. From Tables 3, it can be observed that the BO-LSTM provided lower mean error compared to the manually configured LSTM. However, there was no significant difference between them in terms of both RMSE and MAE. In contrast, BO-GRU gave significantly lower mean error than its hand-tuned version. Moreover, both BO-LSTM and BO-GRU achieved significantly better performance than traditional machine learning models, indicating the ability of BO to configure the models and find hyperparameters values that give similar and sometimes better performance compared to the manually tuned models.

Both BO-LSTM and BO-GRU contributed to the final output of the proposed model, but in a different proportion. In the training phase, BO-LSTM and BO-GRU were used to generate predictions on the training dataset. Afterward, these predictions were reshaped into a suitable format and used to train the blender, which aims to find the appropriate values of w_1 and w_2. The values of w_1 and w_2 for the 30 runs of the Blended-LSTM-GRU were examined. Findings demonstrated that the values of w_1 and w_2 were ranging from 0.69 to 0.91 and from 0.09 to 0.31 respectively, which indicated that the BO-LSTM model had a higher impact on the results of the Blended-LSTM-GRU model.

Forecasting accuracy of the proposed model was visually assessed on an unseen testing data set. Actual demand and forecasting results of BO-LSTM, BO-GRU, and Blended-LSTM-GRU were displayed in Figures 8, 9 and 10 respectively. By analyzing the graphs, it can be observed that these models could provide accurate forecasts and very close predictions to actual observations.

For the hand tuned models, LSTM was the most accurate in terms of RMSE, while FFNN had the lowest error with regards to the MAE, followed by the RF model. RNN and GRU were the worst in terms of both RMSE and

MAE, where it can be observed from Table 3 that shallow and less complicated models such as the RF provided more accurate forecasts. This behavior can be explained by the problem of overfitting, where RNN and GRU have failed to generalize to new unseen dataset.

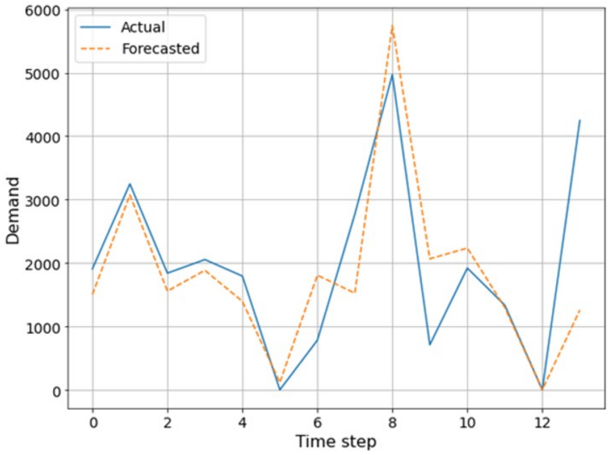

Figure 8. Actual and predicted demand of drinking water bottle packs for 14 days using BO-LSTM model

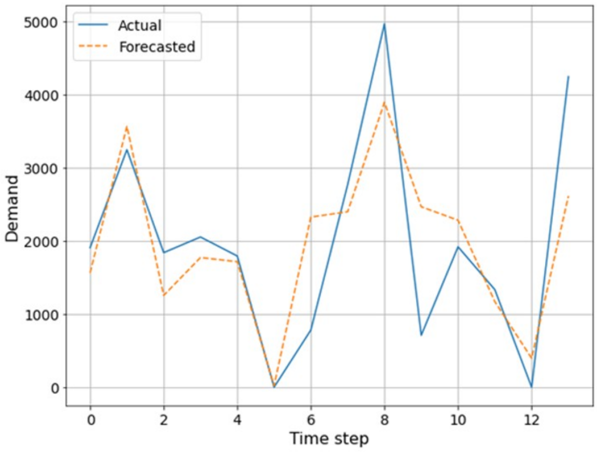

Figure 9. Actual and predicted demand of drinking water bottle packs for 14 days using BO-GRU model

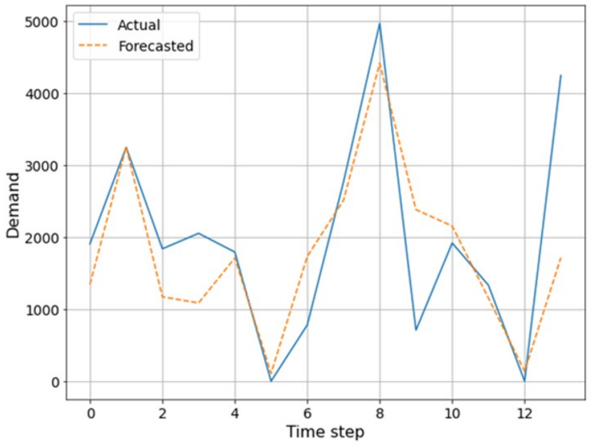

Figure 10. Actual and predicted demand of drinking water bottle packs for 14 days using Blended-LSTM-GRU model

The LSTM model significantly outperforms the GRU model. Further, BO-LSTM resulted in lower error when compared to BO-GRU. Superiority of the LSTM models can be attributed to the complex gating system and its different way of managing the flow of information through it, enabling the LSTM to capture complex time series dynamics.

Generally, finding revealed that the proposed Blended-LSTM-GRU achieved better results than both BO-LSTM and BO-GRU separated, demonstrating the advantage of combining several predictors in an ensemble, where contributors try to account for the relationships left by other members, improving the overall results of the ensemble model. Furthermore, BO-LSTM and BO-GRU models resulted in lower forecasting error than the hand-tuned models, indicating the ability of BO algorithm to effectively configure deep learning models.

With the end of the lockdown period, the market will begin to recover from the effects of the pandemic, and consumers will gradually return to their daily habits as before the pandemic. To confirm the effectiveness of the proposed method and its usability in the normal situation, the Blended-LSTM-GRU model was tested on data collected before the pandemic covers 229 days from July 1, 2019 to February 12, 2020. The data were prepared using the four-step process proposed in section 4.2 where the first 215 data points were used for training and the next 14 points were used for testing. Results showed the effectiveness of the proposed forecasting model where it achieved a mean RMSE and MAE of 972.95 and 804.10 respectively, which is significantly lower than the same model errors when applied to volatile dataset when tested using the t-test. The results show high accuracy when the model is implemented to COVID-19 as a case study. Consequently, it can be concluded that the same model can be utilized in case of disasters such as earthquakes.

6. Conclusion

This study proposed a deep ensemble forecasting method that combines state-of-the-art LSTM and GRU models. Both models were optimized using Bayesian optimization, then aggregated by training a simple model that performs a weighted sum over their predictions. The resulting model which called Blended-LSTM-GRU, was used to forecast drinking water bottle packs demand during the volatile situation of Covid-19 pandemic. The suggested model was statistically compared to its contributors (BO-LSTM and BO-GRU) and other forecasting techniques using the t-test. Results revealed that the Blended-LSTM-GRU model was the most successful with the lowest mean RMSE and MAE, where it was reduced by 2.80 % and 4.74 % compared to BO-LSTM and 3.14 % and 3.60 % compared to BO-GRU respectively. Models optimized using the BO method were able to provide more accurate results than the manually tuned models, indicating the ability of BO to find successful configuration that competes with models constructed by trial and error. Furthermore, graphical examination of BO-LSTM, BO-GRU, and Blended-LSTM-GRU demonstrated the ability of these models to provide forecasts close to the actual data and kept up with the sharp increases and decreases.

As future work, other deep learning architecture such as many-to-many recurrent networks and attention mechanisms will be investigated. Moreover, different aggregation methods of deep learning models for times series forecasting will be tested, and different optimization algorithms will be investigated and compared in time series forecasting applications. Additionally, the relationship between prediction error normality and the prediction model performance will be investigated.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C. et al. (2016). *Tensorflow: Large-scale machine learning on heterogeneous distributed systems.* arXiv preprint arXiv:1603.04467.

Abbasimehr, H., Shabani, M., & Yousefi, M. (2020). An optimized model using LSTM network for demand forecasting. *Computers & Industrial Engineering,* 143, 106435. https://doi.org/10.1016/j.cie.2020.106435

Aday, S., & Aday, M.S. (2020). Impact of COVID-19 on the food supply chain. *Food Quality and Safety,* 4(4), 167-180. https://doi.org/10.1093/fqsafe/fyaa024

Agrawal, S., Jamwal, A., & Gupta, S. (2020). Effect of COVID-19 on the Indian economy and supply chain [Internet]. Preprints. 2020 [cited 2021Dec11]. https://doi.org/10.20944/preprints202005.0148.v1

Akyuz, A.O., Uysal, M., Bulbul, B.A., & Uysal, M.O. (2017). Ensemble approach for time series analysis in demand forecasting: Ensemble learning. In *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)* (7-12). IEEE. https://doi.org/10.1109/INISTA.2017.8001123

Ansari, M.S., Bartos, V., & Lee, B. (2020). Shallow and deep learning approaches for network intrusion alert prediction. *Procedia Computer Science,* 171, 644-653. https://doi.org/10.1016/j.procs.2020.04.070

Ariyo, A.A, Adewumi, A.O., & Ayo, C.K. (2014). Stock price prediction using the ARIMA model. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation* (106-112). IEEE. https://doi.org/10.1109/UKSim.2014.67

Aslam, M., Lee, S.J., Khang, S.H., & Hong, S. (2021). Two-stage attention over LSTM with bayesian optimization for day-ahead solar power forecasting. *IEEE Access,* 9, 107387-107398. https://doi.org/10.1109/ACCESS.2021.3100105

Atalan, A. (2020). Is the lockdown important to prevent the COVID-19 pandemic? Effects on psychology, environment and economy-perspective. *Annals of Medicine and Surgery,* 56, 38-42. https://doi.org/10.1016/j.amsu.2020.06.010

Beam, A.L., Manrai, A.K., & Ghassemi, M. (2020). Challenges to the reproducibility of machine learning models in health care. *Jama,* 323(4),305-306. https://doi.org/10.1001/jama.2019.20866

Bedi, J., & Toshniwal, D. (2019). Deep learning framework to forecast electricity demand. *Applied Energy,* 238, 1312-26. https://doi.org/10.1016/j.apenergy.2019.01.113

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems,* 24.

Bergstra, J., Yamins, D., & Cox, D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. *InInternational Conference on Machine Learning* (115-123). PMLR.

Box, G.E., & Cox, D.R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological),* 26(2), 211-243. https://doi.org/10.1111/j.2517-6161.1964.tb00553.x

Burkov A. (2019). *The hundred-page machine learning book.* Quebec City, QC, Canada: Andriy Burkov.

Carbonneau, R., Laframboise, K., & Vahidov. R. (2008). Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research,* 184(3), 1140-1154. https://doi.org/10.1016/j.ejor.2006.12.004

Chen, P., Yuan, H., & Shu, X. (2008). Forecasting crime using the arima model. In *2008 fifth international conference on fuzzy systems and knowledge discovery* (627-630). IEEE. https://doi.org/10.1109/FSKD.2008.222

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078. https://doi.org/10.3115/v1/D14-1179

Choi, J.Y., & Lee, B. (2018). Combining LSTM network ensemble via adaptive weighting for improved time series forecasting. In *Mathematical Problems in Engineering,* . https://doi.org/10.1155/2018/2470171

Chollet F. (2021). *Deep learning with Python.* Simon and Schuster.

Chowdhury, M., Sarkar, A., Paul, S.K., & Moktadir, M. (2020). A case study on strategies to deal with the impacts of COVID-19 pandemic in the food and beverage industry. *Operations Management Research,* 15, 1-3. https://doi.org/10.1007/s12063-020-00166-9

Chowdhury, P., Paul, S.K., Kaisar, S., & Moktadir, M.A. (2021). COVID-19 pandemic related supply chain studies: A systematic review. *Transportation Research Part E: Logistics and Transportation Review,* 148, 102271. https://doi.org/10.1016/j.tre.2021.102271

D'agostino, R.A., & Pearson, E.S. (1973). Tests for departure from normality. Empirical results for the distributions of b 2 and√ b. *Biometrika,* 60(3), 613-622. https://doi.org/10.1093/biomet/60.3.613

Da Veiga, C.P., Da Veiga, C.R., Catapan, A., Tortato, U., & Da Silva, W.V. (2014). Demand forecasting in food retail: A comparison between the Holt-Winters and ARIMA models. *Wseas Transactions On Business And Economics,* 11(1), 608-614.

Dey, M., & Loewenstein, M.A. (2020). How many workers are employed in sectors directly affected by COVID-19 shutdowns, where do they work, and how much do they earn? *Monthly Labor Review,* 1-9. https://doi.org/10.21916/mlr.2020.6

Dolgui, A., Ivanov, D., & Sokolov, B. (2018). Ripple effect in the supply chain: an analysis and recent literature. *International Journal of Production Research,* 56(1-2), 414-30. https://doi.org/10.1080/00207543.2017.1387680

Fattah, J., Ezzine, L., Aman, Z., El Moussami, H., & Lachhab, A. (2018). Forecasting of demand using ARIMA model. *International Journal of Engineering Business Management,* 10, 1847979018808673. https://doi.org/10.1177/1847979018808673

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* O'Reilly Media, Inc.

Guha, B., & Bandyopadhyay, G. (2016). Gold price forecasting using ARIMA model. *Journal of Advanced Management Science,* 4(2). https://doi.org/10.12720/joams.4.2.117-121

Habtemariam, E., Kekeba, K., Martinez-Ballesteros, M., & Martinez-Alvarez, F. (2023). A Bayesian Optimization-Based LSTM Model for Wind Power Forecasting in the Adama District, Ethiopia. *Energies,* 16(5), 2317. https://doi.org/10.3390/en16052317

Hoda, S., Singh, A., Rao, A., Ural, R., & Hodson, N. (2020). Consumer demand modeling during COVID-19 pandemic. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2282-2289). IEEE. https://doi.org/10.1109/BIBM49941.2020.9313281

Huang, H., Zhang, Z., & Song, F. (2021). An ensemble-learning-based method for short-term water demand forecasting. *Water Resources Management,* 35(6), 1757-1773. https://doi.org/10.1007/s11269-021-02808-4

Ishaq, M., & Kwon, S. (2021). Short-term energy forecasting framework using an ensemble deep learning approach. *IEEE Access,* 9, 94262-94271. https://doi.org/10.1109/ACCESS.2021.3093053

Jain, A., Kumar-Varshney, A., & Chandra-Joshi, U. (2001). Short-term water demand forecast modelling at IIT Kanpur using artificial neural networks. *Water Resources Management,* 15(5), 299-321. https://doi.org/10.1023/A:1014415503476

Jastrzębski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y. et al. (2018). Width of minima reached by stochastic gradient descent is influenced by learning rate to batch size ratio. In *International Conference on Artificial Neural Networks* (392-402). Springer, Cham. https://doi.org/10.1007/978-3-030-01424-7_39

Jin, X.B., Zheng, W.Z., Kong, J.L., Wang, X.Y., Bai, Y.T., Su, T.L. et al. (2021). Deep-learning forecasting method for electric power load via attention-based encoder-decoder with bayesian optimization. *Energies,* 14(6), 1596. https://doi.org/10.3390/en14061596

Jin, Y., Ye, X., Ye, Q., Wang, T., Cheng, J., & Yan, X. (2020). Demand forecasting of online car-hailing with stacking ensemble learning approach and large-scale datasets. *IEEE Access,* 8, 199513-199522. https://doi.org/10.1109/ACCESS.2020.3034355

Kamal, I.M., Bae, H., Sunghyun, S., & Yun, H. (2020). DERN: Deep ensemble learning model for short-and long-term prediction of Baltic dry index. *Applied Sciences,* 10(4), 1504. https://doi.org/10.3390/app10041504

Kandel, I., & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express,* 6(4), 312-315. https://doi.org/10.1016/j.icte.2020.04.010

Kantasa-Ard, A., Bekrar, A., & Sallez, Y. (2019). Artificial intelligence for forecasting in supply chain management: A case study of White Sugar consumption rate in Thailand. *IFAC-PapersOnLine,* 52(13), 725-730. https://doi.org/10.1016/j.ifacol.2019.11.201

Khashei, M., & Bijari, M. (2011). A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. *Applied Soft Computing,* 11(2), 2664-2675. https://doi.org/10.1016/j.asoc.2010.10.015

Kofinas, D., Mellios, N., Papageorgiou, E., & Laspidou. C. (2014). Urban water demand forecasting for the island of Skiathos. *Procedia Engineering,* 89, 1023-1030. https://doi.org/10.1016/j.proeng.2014.11.220

Lahouar, A., & Slama, J.B. (2015). Day-ahead load forecast using random forest and expert input selection. *Energy Conversion and Management,* 103, 1040-1051. https://doi.org/10.1016/j.enconman.2015.07.041

Liang, X., Lin, Y., Deng, C., Mo, Y., Lu, B., Yang, J. et al. (2024). Machine Learning-Based Sales Prediction Using Bayesian Optimized XGBoost Algorithms. *Modern Management Based on Big Data V,* 387, 248-268. https://doi.org/10.3233/FAIA240263

Micheal, N.E., Hasan, S., Al-Durra, A., & Mishra, M. (2022). Short-term solar irradiance forecasting based on a novel Bayesian optimized deep Long Short-Term Memory neural network. *Applied Energy,* 324, 119727. https://doi.org/10.1016/j.apenergy.2022.119727

Minondo, A. (2021). Impact of COVID-19 on the trade of goods and services in Spain. *Applied Economic Analysis,* 29(85), 58-76. https://doi.org/10.1108/AEA-11-2020-0156

Ozaki, S,. Ooka, R., & Ikeda, S. (2021). Energy demand prediction with machine learning supported by auto-tuning: a case study. *Journal of Physics, Conference Series,* 2069, 012143. https://doi.org/10.1088/1742-6596/2069/1/012143

Pai, P.F., & Lin, C.S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega,* 33(6), 497-505. https://doi.org/10.1016/j.omega.2004.07.024

Parmezan, A.R., Souza, V.M., & Batista, G.E. (2019). Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information Sciences,* 484, 302-337. https://doi.org/10.1016/j.ins.2019.01.076

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O. et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research,* 12, 2825-2830.

Pichler, A., & Farmer, J.D. (2021). Simultaneous supply and demand constraints in input-output networks: The case of Covid-19 in Germany, Italy, and Spain. *Economic Systems Research,* 34(3), 273-293. https://doi.org/10.1080/09535314.2021.1926934

Qiu, X., Ren, Y., Suganthan, P.N., & Amaratunga, G.A. (2017). Empirical mode decomposition based ensemble deep learning for load demand time series forecasting. *Applied Soft Computing,* 54, 246-255. https://doi.org/10.1016/j.asoc.2017.01.015

Qiu, X., Zhang, L., Ren, Y., Suganthan, P.N., & Amaratunga, G. (2014). Ensemble deep learning for regression and time series forecasting. In *2014 IEEE symposium on computational intelligence in ensemble learning (CIEL)* (1-6). IEEE. https://doi.org/10.1109/CIEL.2014.7015739

Raouf, M., Elsabbagh, D., & Wiebelt, M. (2020). *Impact of COVID-19 on the Jordanian economy: Economic sectors, food systems, and households.* International Food Policy Research Institute (IFPRI). https://doi.org/10.2499/p15738coll2.134132

Roggeveen, A.L., & Sethuraman, R. (2020). How the COVID-19 pandemic may change the world of retailing. *Journal of Retailing,* 96(2), 169. https://doi.org/10.1016/j.jretai.2020.04.002

Saarinen, L., Loikkanen, L., Tanskanen, K., Kaipia, R., Takkunen, S., & Holmström, J. (2020). *Agile planning: Avoiding disaster in the grocery supply chain during COVID-19 crisis.* Aalto University.

Sahoo, B.B., Jha, R., Singh, A., & Kumar, D. (2019). Long short-term memory (LSTM) recurrent neural network for low-flow hydrological time series forecasting. *Acta Geophysica,* 67(5):1471-1481. https://doi.org/10.1007/s11600-019-00330-1

Shereen, M.A., Khan, S., Kazmi, A., Bashir, N., & Siddique. R. (2020). COVID-19 infection: Emergence, transmission, and characteristics of human coronaviruses. *Journal of Advanced Research,* 24, 91-98. https://doi.org/10.1016/j.jare.2020.03.005

Sheth, J. (2020). Impact of Covid-19 on consumer behavior: Will the old habits return or die? *Journal of Business Research,* 117, 280-283. https://doi.org/10.1016/j.jbusres.2020.05.059

Siami-Namini, S., Tavakoli, N., & Namin, A.S. (2018). A comparison of ARIMA and LSTM in forecasting time series. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)* (1394-1401). IEEE. https://doi.org/10.1109/ICMLA.2018.00227

Smolak, K., Kasieczka, B., Fialkiewicz, W., Rohm, W., Siła-Nowicka, K., & Kopańczyk, K. (2020). Applying human mobility and water consumption data for short-term water demand forecasting using classical and machine learning models. *Urban Water Journal,* 17(1), 32-42. https://doi.org/10.1080/1573062X.2020.1734947

Spiliotis, E., Makridakis, S., Semenoglou, A.A., & Assimakopoulos, V. (2020). Comparison of statistical and machine learning methods for daily SKU demand forecasting. *Operational Research,* 22, 3037-3061 .

Tan, M., Yuan, S., Li, S., Su, Y., Li, H., & He, F. (2019). Ultra-short-term industrial power demand forecasting using LSTM based hybrid ensemble learning. *IEEE Transactions on Power Systems,* 35(4), 2937-2948. https://doi.org/10.1109/TPWRS.2019.2963109

Tugay, R., & Oguducu, S.G. (2020). Demand prediction using machine learning methods and stacked generalization. arXiv preprint arXiv:2009.09756.

Usmani, M., Memon, Z., Danyaro, K., & Qureshi, R. (2024). Optimized Multi-Level Multi-Type Ensemble (OMME) Forecasting Model for Univariate Time Series. *IEEE Access,* 12, 35700-35715. https://doi.org/10.1109/ACCESS.2024.3370679

Wen, L., Zhou, K., & Yang, S. (2020). Load demand forecasting of residential buildings using a deep learning model. *Electric Power Systems Research,* 179, 106073. https://doi.org/10.1016/j.epsr.2019.106073

World Health Organization (2021). Coronavirus disease (Covid-19) [Internet]. Available at: https://www.who.int/health-topics/coronavirus#tab=tab_1

Xenochristou, M., & Kapelan, Z. (2020). An ensemble stacked model with bias correction for improved water demand forecasting. *Urban Water Journal,* 17(3), 212-223. https://doi.org/10.1080/1573062X.2020.1758164

Zeroual, A., Harrou, F., Dairi, A., & Sun, Y. (2020). Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study. *Chaos, Solitons & Fractals,* 140, 110121. https://doi.org/10.1016/j.chaos.2020.110121

Zhang, G.P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing,* 50, 159-175. https://doi.org/10.1016/S0925-2312(01)00702-0

Journal of Industrial Engineering and Management, 2025 (www.jiem.org)