Price Forecasting in the Ontario Electricity Market via TriConvGRU Hybrid Model: Univariate vs. Multivariate Frameworks

par

Behdad Ehsani

Pierre-Olivier Pineau

HEC Montréal

Directeur de recherche

Laurent Charlin

HEC Montréal

Codirecteur de recherche

Mémoire présenté en vue de l'obtention

du grade de maîtrise ès science des données et analytique d'affaires

Département de sciences de la décision

Janvier 2023

Résumé

La prévision des prix de l'électricité est une tâche difficile pour les décideurs sur les marchés de l'électricité déréglementés en raison des caractéristiques inhérentes aux prix de l'électricité, par exemple, la fréquence élevée et la volatilité. Par conséquent, une prévision précise des prix de l'électricité peut aider les participants au marché à maximiser leurs bénéfices. En conséquence, nous avons proposé un nouveau modèle hybride d'apprentissage profond pour prévoir les prix de l'électricité en Ontario à un, deux et trois pas à l'avance, basé sur un réseau neuronal convolutif (CNN) et une unité récurrente à grille (GRU). Notre modèle se compose de trois modèles CNN-GRU consécutifs combinés en parallèle avec différentes données d'entrée. Nous avons sous-échantillonné les données d'entrée via des couches de mise en commun au début de deux flux du modèle afin de capturer simultanément différentes fréquences de modèles de prix. En outre, un ensemble de variables externes, y compris les prix précédents, la charge électrique, la production, l'importation et l'exportation, et les données météorologiques, ont été prises en compte dans nos modèles de prévision pour vérifier si ces caractéristiques améliorent l'efficacité des modèles. Enfin, trois expériences portant sur différentes semaines de 2022 ont été réalisées sur le marché de l'électricité de l'Ontario afin d'évaluer le modèle proposé. Les résultats montrent que le modèle proposé a réduit de manière significative l'erreur de prévision de 63,3 % dans la première expérience, de 41,8 % dans la deuxième et de 28,22 % dans la troisième, en moyenne. En outre, le modèle proposé a été comparé à plusieurs modèles de base, notamment des modèles statistiques de séries chronologiques, d'apprentissage automatique et d'apprentissage profond. En outre, la comparaison des résultats dans des contextes univariés et multivariés a indiqué que l'ajout de variables aux modèles de prévision ne permettait pas de réduire les erreurs de prévision.

Mots-clés: Prévision des Prix de l'électricité, Apprentissage Profond, Apprentissage Automatique, Modèle Hybride, Marché de l'électricité de l'Ontario

Abstract

Electricity price forecasting is a challenging task for decision-makers in deregulated power markets due to the inherent characteristics of electricity prices, e.g., high frequency and volatility. Therefore, accurate forecasting of electricity prices can assist market participants in maximizing their profit. Accordingly, we proposed a novel hybrid Deep Learning model to forecast one-step, two-step, and three-step ahead Ontario electricity prices based on a Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU). Our model consists of three consecutive CNN-GRU models combined in parallel with different input data. We downsampled input data via pooling layers at the beginning of two streams of the model to capture different frequencies of price patterns concurrently. Also, a set of external variables, including previous prices, electricity load, generation, import and export, and weather data, were considered in our forecasting models to test whether these features improve the efficiency of the models. Finally, three experiments in various weeks of 2022 were carried out in the Ontario electricity market to assess the model. The results indicate that the proposed model reduced the forecasting error significantly by 63.3% in the first experiment, 41.8% in the second, and 28.2% in the third, on average, with respect to a Root Mean Square Error (RMSE). Also, the proposed model was compared with outperformed several baseline models, including statistical time-series, Machine Learning, and Deep Learning models. Furthermore, the comparison of results in univariate and multivariate settings indicated that adding variables to forecasting models did not help reduce forecasting errors.

Keywords: Electricity Price Forecasting, Deep Learning, Machine Learning, Hybrid Model, Ontario Electricity Market

Acknowledgements

I extend my sincerest gratitude to my supervisor, Professor Pierre-Olivier Pineau, for his comprehensive knowledge of energy markets, which enabled me to choose an interesting topic for my thesis and overcome the challenges I encountered while writing it. I also appreciate his benevolence, patience, support, and encouragement throughout the process.

I am also grateful to my co-supervisor, Professor Laurent Charlin, for the wealth of knowledge and insights he imparted through his Machine Learning course materials. His materials exposed me to the various applications of machine learning and state-of-the-art methods in various fields. Furthermore, I appreciate his constructive feedback which helped me to improve my research skills in the field of AI.

I would like to express my appreciation to Mr. Nima Rafizadeh for his expertise in Data Science and Energy markets. His guidance was instrumental in implementing forecasting methods, a crucial aspect of my project, and it would have been much more difficult without his assistance. I also gained valuable insights through our discussions on how to tackle research problems and devise new methods.

Lastly, I would like to extend my gratitude to my family, particularly my parents, for their unwavering support throughout my research journey and for providing emotional support during this challenging year.

Contents

List of Figures

List of Tables

1. Introduction

In all competitive markets, prices are defined by supply and demand. Over decades, electricity markets globally are under reformation which resulted in deregulated markets. This has led to the establishment of electricity power exchanges, e.g., independent electricity system operator (IESO) and Alberta electric system operator (AESO) in Canada; Pennsylvania-New Jersey-Maryland interconnection (PJM) and New York independent system operator (NYISO) in the USA; and the National Electricity Market (NEM) in Australia, enabling competitive trading in the electricity market (Maciejowska et al., 2021). The electricity market differs from other commodity markets because both the demand and supply sides are more complex than in most markets. On the demand side, consumers have a relatively inelastic demand, which follows time patterns. Therefore, the demand must be carefully balanced in order to prevent blackouts or overloading of the network infrastructure. On the supply side, a combination of inflexible, flexible, intermittent, and potentially continuous electricity generation facilities are used, leading to an increase in the complexity of the energy supply (Lehna et al., 2022).

Furthermore, despite recent developments in storage, substantial quantities of electricity cannot be stored at a reasonable price in contrast to other commodities, e.g., natural gas and oil (Park et al., 2006). The lack of resources for storing electricity makes prices highly sensitive to shocks in demand and supply (Ioannidis et al., 2021). Notably, some shocks result from power plant outages and instability in the electricity generation of power plants. Therefore, demand and supply shocks result in the phenomena of negative prices (Atănăsoae et al., 2020) and extremely high prices, called spikes, which do not usually occur in other commodity markets. For instance, although negative prices followed a decreasing trend in North American electricity markets from 2016 to 2020, negative prices were still a sensible percentage of the hourly price (Rafizadeh, 2022), as shown in Figure 1.

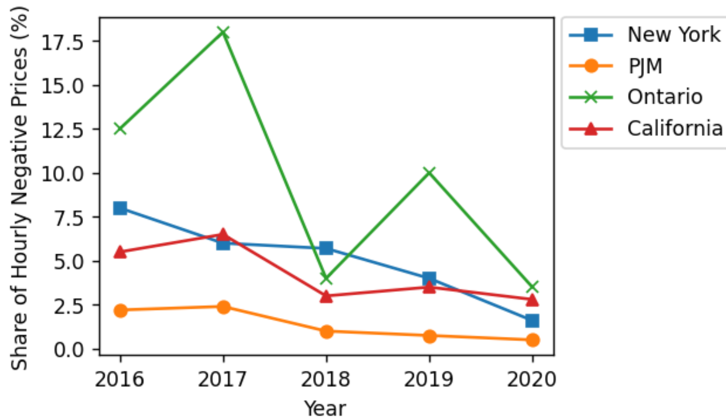


Figure 1: Percentage of hours with negative prices in North American electricity markets

Nowadays, prediction and analysis based on historical data play a critical role in energy decision-making. In electricity markets, data analyses help decision-makers tackle several issues, including *system reliability* by predicting equipment failures and maintenance needs, *price forecasting* by helping market participants in trades, *demand forecasting* by ensuring that the generated electricity can meet the demand, and *market monitoring* by finding market manipulations (vom Scheidt et al., 2020).

In terms of electricity price forecasting, it can help energy decision-makers balance supply and demand throughout the day to control the sporadic collapse in prices, decrease the occurrence of negative prices, and maintain the electric grid reliably (Benini et al., 2002). In addition, from the supplier's side, generators can benefit from the price forecast, illustrating the price trend in subsequent hours, by optimizing their offering strategy (Pourdaryaei et al., 2019). Concerning the demand side, the price projections help energy traders find lucrative opportunities (Dagoumas et al., 2017). To be specific, some consumers, e.g., dispatchable loads, in the market use the forecast to find the future price trend, playing a pivotal role in their bidding strategy in the auction.

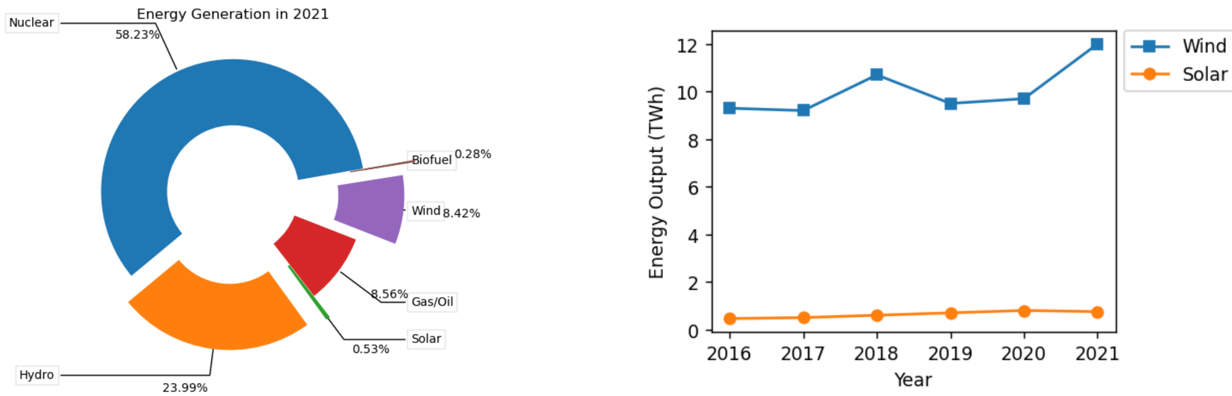
Regarding the electricity price forecast, there are two major reasons complicating it in both the long and short terms (Yang and Schell, 2022b), elaborated in the following:

Abbreviation			
AESO	Alberta electric system operator	*LR*	linear regression
ANN	artificial neural network	*LSTM*	long short-term memory
AR	autoregressive	*MAE*	mean absolute error
ARIMA	autoregressive integrated moving average	*MCP*	market clearing prices
Att	attention mechanism	*ML*	machine learning
CNN	convolutional neural network	*MSE*	mean squared error
DL	deep learning	*NEM*	National Electricity Market
DT	decision tree	*NYISO*	New York independent system operator
EPF	electricity price forecasting	*PCC*	Pearson correlation coefficient
GARCH	generalized autoregressive conditional heteroskedasticity	*PJM*	Pennsylvania-New Jersey-Maryland Interconnection
GHG	greenhouse gases	*ReLU*	rectified linear unit
GRU	gated recurrent unit	*RF*	random forest
HO	hyper-parameter optimization	*RNN*	recurrent neural network
HOEP	hourly Ontario energy price	*SVR*	support vector regression
IESO	independent electricity system operator	*VAR*	vector autoregression
KNN	k-nearest neighbors	*XGB*	extreme gradient boosting

⋆ Comprehensive List of Abbreviations with their Full Forms in the Article

1. *Uncertainty*: Temporal variability in the electricity market causes uncertainty in prices. The effective factors can be grouped into two categories: technical issues, e.g., grid outages, and economic uncertainties, e.g., fluctuations in fuel prices (Ehsan and Yang, 2019).

2. *Volatility*: The volatility of prices is another major challenge in real-time price forecasting. As discussed before, price spikes and negative prices occur sporadically due to the demand and supply shocks inherent in the electricity market, making the market volatile. Furthermore, regarding renewable energies aiming at reducing greenhouse gas (GHG) emissions (Liu et al., 2011), they affect supply offers and have a fundamental effect on electricity price distribution (Mulder and Scholtens, 2013; Rintamäki et al., 2017), including average and standard deviation throughout time. For example, as shown in Figure 2, almost 33% of the electricity in the Ontario electricity market was generated by renewable energy, and the trend of electricity generated by solar and wind plants increases gradually, leading to more price volatility.



(a) Percentage of each generators in 2021

(b) Generation trend of electricity by wind and solar

Figure 2: Renewable energy generation in Ontario electricity market

Regarding the aforementioned challenges in electricity price forecasting and the significance of price forecasting for market participants, electricity price models are required for the reliable operation of the power grid (Zarnikau et al., 2019). As the main contribution of this work, we propose a novel Deep Learning (DL) model that can capture different price patterns with its multiple streams, to forecast the electricity prices in the Ontario electricity market and compare it with commonly used Machine Learning (ML) and statistical models. Also, the effect of external variables, particularly renewable energies, as a driving force of price volatility was considered in our research.

1.1. Background

1.1.1. Background in Ontario Electricity Market

There are two separate pricing mechanisms in deregulated electricity markets. The first, known as the intra-day market, allows for the buying and selling of electricity on the same trading day (Terlouw et al., 2019). The second mechanism, known as the day-ahead market, establishes prices for next-day delivery through a day-ahead auction exchange (Lehna et al., 2022). In terms of the Ontario electricity market, the electricity is traded through the intra-day electricity market, and the real-time prices are determined by the following steps (IESO, 2014). Note that market participants in the Ontario electricity market are dispatchable and non-dispatchable.

First, dispatchable suppliers, including nuclear, large natural gas, hydroelectric, and coal-fired facilities, and dispatchable consumers, e.g., some large consumers, send their offers and bids to provide enough electricity to meet Ontario energy needs every five minutes. Simultaneously, non-dispatchable suppliers, e.g., self-scheduling and intermittent generators, and non-dispatchable consumers, e.g., local distribution companies (LDC), submit their schedules for electricity generation and consumption at each time step, respectively, and cannot submit any bids or offers in the market. The supplier's offers indicate the amount of electricity they can produce and at what price, whereas the consumer's bids indicate the amount of electricity they need and at what price. Afterwards, the offers are reviewed and arranged in ascending order according to the prices obtained through the auction. Second, non-dispatchable electricity, including the electricity generated by non-dispatchable suppliers and consumed by non-dispatchable customers, is allocated at the beginning. Third, the bids and offers are matched, starting with the lowest-cost options, until enough energy is secured to meet the dispatchable consumer's needs.

The process sets a new market clearing price (MCP) every five minutes. Thereafter, an hourly Ontario energy price (HOEP) is computed by taking the weighted average of the twelve MCPs that are established during each hour.[1] HOEP is considered the target variable for forecasting in our research because it depicts the price trend, which is useful for dispatchable participants in their bidding and offering strategies. The IESO provides one-hour, two-hour, and three-hour ahead price forecasts to help market participants optimize their trading strategies.[2] Hence, our study uses a forecasting horizon of three steps, which aligns with the range of forecasts provided by the IESO.

Aside from the pricing mechanism in the Ontario market, there are two types of demand in the market: *market demand* which is calculated by adding all output from registered generators in the market and all scheduled imports to the province, and *Ontario demand*

[1] https://www.ieso.ca/power-data/price-overview/hourly-ontario-energy-price

[2] http://reports.ieso.ca/public/PriceHOEPPredispOR/

which is calculated by subtracting scheduled exports from market demand.[3]

1.1.2. Background in Forecasting Models

In this section, a brief description of the base forecasting models used in the electricity market is presented.

AutoRegressive (AR): In time-series forecasting, an AR method is a prevalent technique that models the variable by a linear formulation of the historical values of the variable. The basic idea behind this method is that there is a pattern in the data that can be captured using a set of parameters that can be used to forecast future values (De Alba, 1993). The basic equation for an AR model is represented as:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t \tag{1}$$

where y_t represents the price at time t, c represents a constant, $\phi_1, \phi_2, \cdots, \phi_p$ are coefficients, and ϵ_t represents white noise term. The p is referred to as the order of the AR model. Note that the AR coefficients are generally computed using the least squares technique, which minimizes the sum of the squared differences between the real and forecasted values.

AutoRegressive Integrated Moving Average (ARIMA): An ARIMA method is another statistical model commonly used for forecasting time series data. It builds upon the principles of both the AR and moving average (MA) methods, integrating them to produce a more effective forecasting tool. The fundamental concept behind ARIMA is to model the differences between consecutive values in a time series instead of the actual values. This is referred to as the *integrated* part of the model, and it helps to make the time series stationary (Zhang, 2003). The AR and MA components of the model are employed to identify patterns in differences. The basic equation for an ARIMA model is represented as:

$$y'_t = c + \phi_1 y'_{t-1} + \phi_2 y'_{t-2} + ... + \phi_p y'_{t-p} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q} + \epsilon_t \tag{2}$$

[3] https://www.ieso.ca/en/Power-Data/Demand-Overview/Real-time-Demand-Reports

where y'_t represents the differenced value at time t, c represents a constant, $\phi_1, \phi_2, \cdots, \phi_p$ are coefficients of the AR part, $\theta_1, \theta_2, \cdots, \theta_q$ are the moving average coefficients, and ϵ_t represents a white noise term. The values of p, d, and q are referred to as the order of the ARIMA model, where p, d, and q represent the order of the AR part, integrated component, and the MA component, respectively. Specifically, p represents the number of past time steps that are used as predictors in the model, d represents the number of times the data has been differenced to make it stationary, and q represents the number of lagged forecast errors that are used as predictors in the model

Vector AutoRegressive (VAR): Another statistical model, called a VAR model, is an improvement of the univariate AR that can be used to model multiple related time series. The fundamental principle of a VAR model is that each time series is expressed as a linear combination of the historical values of all other time series in the model. This allows the model to learn the interdependencies between several time series (Canova, 1999). The basic equation for an AR model is represented as:

$$y_t = A_1 y_{t-1} + A_2 y_{t-2} + \cdots + A_p y_{t-p} + e_t, \tag{3}$$

where y_t is a vector of current values for all time series, $A_1, A_2, ..., A_p$ represent the VAR coefficients and e_t is a vector of error terms. Similar to the AR model, the number of lags (p), which defines how far back in past values the model will consider, needs to be determined.

Linear Regression (LR): An LR assesses the impact of independent variables on a dependent variable. The ordinary least squares method is employed to estimate the relationship between variables.

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon_t \tag{4}$$

where y is the forecasted price, $X_1, X_2, \cdots, X_n$ are independent variables, $\beta_1, \beta_2, \cdots, \beta_n$ are the estimated coefficients of LR, and ϵ_t is an error term.

The ordinary Least Squares (OLS) method is typically utilized to determine the optimal coefficient values. The OLS method minimizes the sum of squared differences between actual and predicted values of the dependent variable.

Support Vector Regression (SVR): An SVR is a variation of the Support Vector Machine (SVM) algorithm, which is typically utilized for solving classification problems. SVR works by determining the optimum decision function (regression line or curve) to suit the provided data points while balancing model complexity and error tolerance. The fundamental concept is to establish a margin around the decision function, such that most of the data points fall inside this margin (Smola and Schölkopf, 2004). The objective is to decrease the model complexity by maximizing the margin width while penalizing data points that are outside the margin.

K-Nearest Neighborhood (KNN): A KNN algorithm is a non-parametric ML model that can be applied to the task of regression. The concept of prediction is based on the KNN, which first finds the K-nearest data points within the feature space using a distance metric such as Euclidean distance and then averages the target variable values (Cover, 1968). The number of nearest neighbors (K) is an important hyperparameter that should be defined first. The value of K greatly influences the forecasting result. A larger K leads to a smoother, more generalized forecast, whereas a smaller K produces a more complex forecast that is sensitive to outliers. Aside from the high interpretability and simplicity of the KNN model, it can be affected by the curse of dimensionality, and the performance of KNN algorithms decreases as the number of feature variables increases (Marimont and Shapiro, 1979).

Decision Tree (DT): A DT is a technique that can be applied to both regression and classification problems. The data is divided into smaller subgroups in the DT based on the values of the input features. It should be noted that the process of creating a decision tree entails identifying the optimal feature and threshold for splitting the data at each node. Data splitting starts at the root node and progresses via a branched tree to a leaf node that represents the forecasts or the final outcomes (Quinlan, 1996).

Random Forest (RF): Similar to DT, an RF algorithm is a model that is utilized for both classification and regression problems. The RF is based on an ensemble method that utilizes multiple decision trees to make predictions (Abellán et al., 2017). In other words, RF comprises a set of DTs, each of which is trained on a randomly selected bunch of the data. Finally, each tree in the RF algorithm generates a prediction, and the final prediction is calculated by taking the average of the predictions generated by decision trees. The utilization of predictions from multiple trees helps mitigate the overfitting problem commonly encountered in the DT model (Li et al., 2018).

Artificial Neural Network (ANN): An ANN, called a Neural Network (NN) alternatively, is an ML model that seeks to replicate the structure of the human brain. The model is designed to process and transmit information and is composed of layers of interconnected neurons (Divina et al., 2019). Each neuron within the model receives input from other neurons. Then, it performs a mathematical calculation on that input. Finally, it transmits the output to other neurons within the next layer. This process continues until the final output is generated. Note that the final output is dependent on the activation function utilized within each neuron, which can capture non-linearity.

One of the common types of ANN is the feedforward neural network, in which the flow of information from the input layer to the output layer is unidirectional with no feedback loops (Gardner and Dorling, 1998). Feedforward neural networks comprise several layers of artificial neurons, called hidden layers, as shown in Figure 3. Similar to the ANN architecture, the input is fed to several layers of neurons, called hidden layers, before reaching the output layer. These layers process the input data by applying a set of learned parameters, including weights and biases. The weights and biases are learned during the training process by minimizing a loss function between the forecasted value and the actual one.

Recurrent Neural Networks (RNN): An RNN is a type of ANN that is designed to process sequential data. It is called recurrent because it utilizes sequential information by incorporating the data and hidden state from the previous time step in its processing (Sutskever et al., 2011). As shown in Figure 4, RNNs consist of a series of interconnected units, which are organized into layers. Each unit receives input from the data at the current

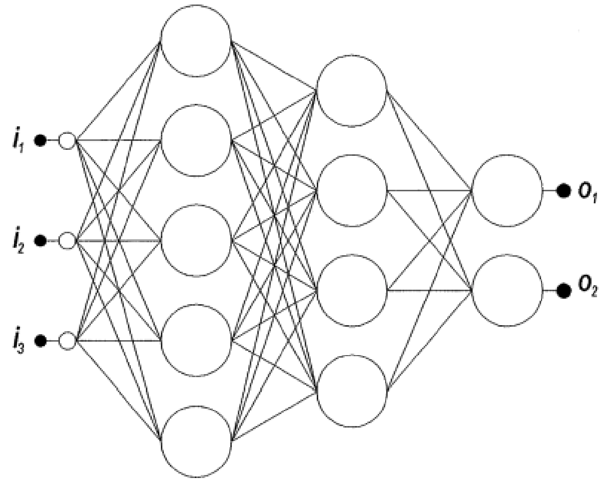


Figure 3: Architecture of an ANN network with two hidden layers (Gardner and Dorling, 1998) $I = [i_1, i_2, i_3]$ is an input vector and $O = [O_1, O_2]$ is an output vector

time step and the hidden state at the previous time step (Xia et al., 2018) and generates an output and a new hidden state. The hidden state represents past information that the RNN has processed and functions as a type of memory within the network. As seen in Figure 5, the process of generation of the hidden state and the output at each time step is as follows:

$$h_t = \tanh(W_h\,[h_{t-1}, x_t] + b_h) \tag{5}$$

$$y_t = \tanh(W_y h_t + b_y) \tag{6}$$

where x_t is the input data at time t; W_h and W_y are the weight matrices for the hidden state and the output, b_h and b_y are the bias terms for the hidden state and the output; h_t is the hidden state at time t; and *tanh* is the hyperbolic tangent activation function.

A Long Short-Term Memory (LSTM) network is a specific type of RNN that can learn long-term dependencies, proposed by Hochreiter and Schmidhuber (1997). One key difference between LSTMs and traditional RNNs is the inclusion of gates within LSTMs to control the flow of information, as shown in Figure 6. An LSTM network has three gates: an *input* gate, an *output* gate, and a *forget* gate. The input gate controls the flow of data into the cell,

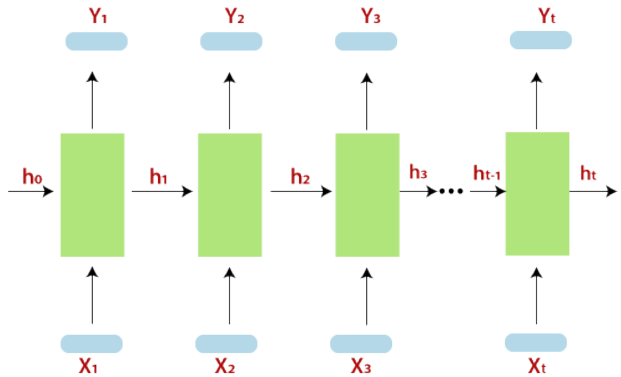


Figure 4: A sequence of RNNs
(Source: https://www.javatpoint.com/training-of-rnn-in-tensorflow/)

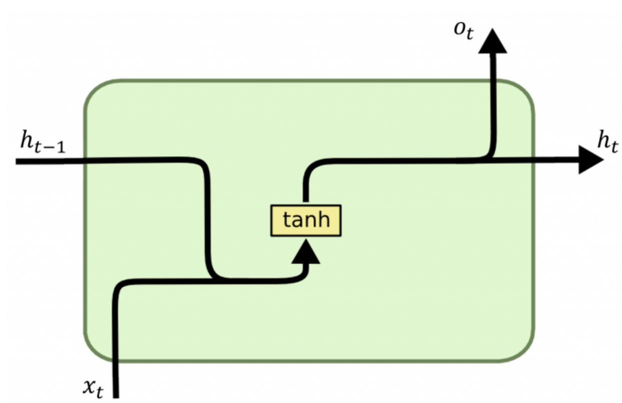


Figure 5: Architecture of an RNN
(Source: http://dprogrammer.org/rnn-lstm-gru/)

the output gate regulates the flow of data out of the cell and back into the network, and the forget gate manages the flow of data from the previous state of the cell into the current state (Salman et al., 2018). Gates are neural networks that control the flow of information through a sequential chain. In other words, these gates allow LSTMs to selectively retain or discard information, which helps to avoid the vanishing gradient problem that can occur in the vanilla RNN (Le et al., 2015). Thus, LSTMs are more effective at handling long-term dependencies than traditional RNNs, but they also require more computational resources and can be more challenging to train.

A gated recurrent unit (GRU) was proposed by Chung et al. (2014) to reduce the computational complexity. Similar to the LSTM, it addresses the issue of the vanishing gradient, which arises when gradient propagation fails in long-term dependencies. However, the GRU has fewer parameters compared to the LSTM. In the GRU architecture, two gates including a reset gate and an update gate, as illustrated in Figure 7, are utilized for information transfer, in contrast to the LSTM architecture that uses three gates (Hossain et al., 2021b). Thus, the GRU needs fewer trainable parameters, resulting in training the model in a shorter time period in comparison to LSTM (Elsayed et al., 2018). Eqs. 7 and 8 represent the mathematical formulation of update and reset gates. The reset gate in a GRU determines which information from the previous time step to keep and which to discard, while the update gate determines what information to store in the current time step. The final output and hidden state of a GRU at time t are obtained by combining the output of the reset and update gates with the input data and hidden state from the previous time step, expressed in Eqs. 9 and 10. In the formulations, the hidden state at time t is based on the previous hidden state and the candidate hidden state, where the contribution of the candidate hidden state is controlled by the update gate. The candidate hidden state is derived by combining the previous hidden state, which is weighted by the reset gate, and the input data.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{7}$$

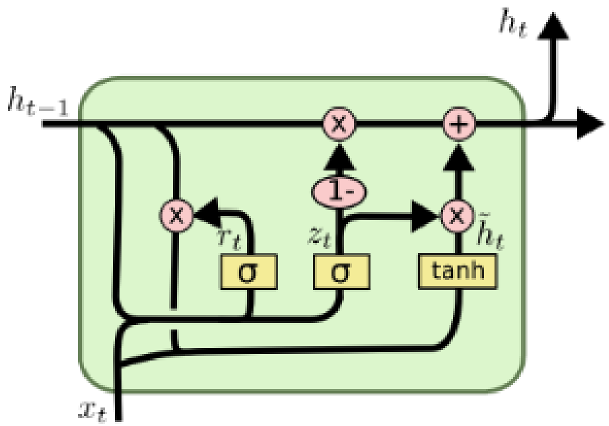


Figure 6: Architecture of LSTM network
(Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

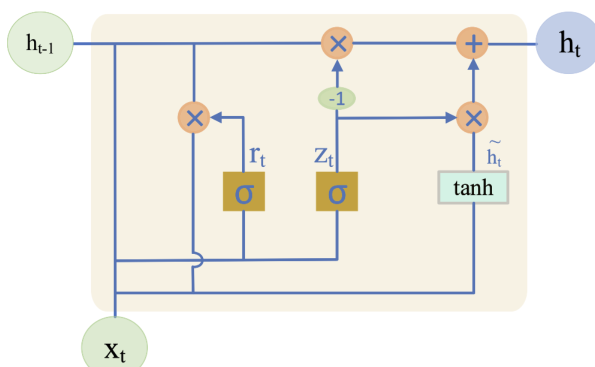


Figure 7: Architecture of GRU network (Hossain et al., 2021b)

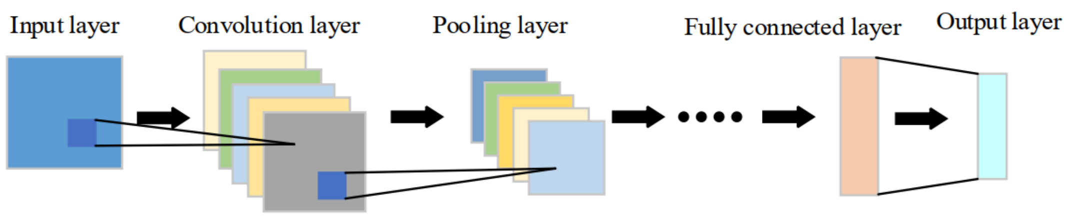


Figure 8: The architecture of the CNN network (Zhao et al., 2019)

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{8}$$

$$\hat{h_t} = tanh(W \cdot [r_t \cdot h_{t-1}, x_t] + b) \tag{9}$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \hat{h_t} \tag{10}$$

where r_t is the reset gate at time t; z_t is the update gate at time t; h_t is the hidden state at time t; x_t is the input data at time t; W_r and W_z are the weight matrices for the reset and update gates; b_r, b_z and b are the bias vectors for the reset and update gates; $\hat{h_t}$ is the candidate hidden state at time t; and σ is the sigmoid and $tanh$ is the hyperbolic tangent activation function.

Convolutional Neural Network (CNN): A CNN is typically designed for grid data to identify all local relations that are invariant among different dimensions (LeCun et al., 1998). CNN can also be applied to one-dimensional data, e.g., time-series data. In the context of a time-series study, a one-dimensional CNN can be employed to extract important features of non-linear relationships in electricity prices. A typical CNN consists of convolution layers, pooling layers, and fully connected layers, and it is capable of identifying complex patterns in data (Hossain et al., 2021b). The structure of a typical CNN is illustrated in Figure 8.

The convolution layer, which is the first step in the process, receives the input data from the data stream and applies its kernel function to extract features from the input data, by sliding a kernel (filter) over it, as illustrated in Figure 9. Mathematically speaking, the convolution layer employs a mathematical convolution operator, rather than matrix multiplication (Hossain et al., 2021a). The convolution operation is defined as follows:

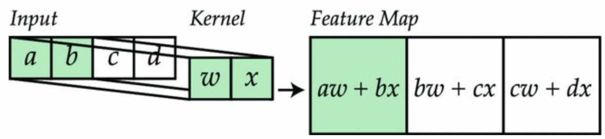


Figure 9: The illustration of a kernel operation (Baldominos et al., 2018)

$$c_i^l = f\left(\sum_{i=1}^{n} x_i * w_i^l + b_j\right) \tag{11}$$

where c_i^l is the ith element of lth feature graph in the convolution layer determined by convolution operation, x_i the ith element of the input graph, w_i^l the ith element of lth convolution kernel, b_j a bias, $*$ convolution operation and f an activation function, e.g., rectified linear unit (ReLU), as the highly used activation function in the CNN structure because it can significantly accelerate the training process and improve the model performance. The ReLU activation function is defined as $f(x) = max(0, x)$, which means that the output of the ReLU function is the maximum value between 0 or the input value x (Agarap, 2018).

In the second step, pooling layers are another key component of convolutional neural networks. They reduce the spatial dimensions of the feature maps produced by convolution layers through a process called down-sampling. This enhances the efficiency of the CNN by decreasing the computational requirements of subsequent layers and helps prevent overfitting by requiring the model to utilize more general and abstract features instead of highly specific ones that may only be present in the training data. Two commonly used types of pooling are *max pooling* and *average pooling*.[4] In max pooling, the maximum value within a small spatial window is selected and the remaining values are discarded, resulting in a lower-resolution output with increased contrast. In contrast, average pooling calculates the average of all values within a small spatial window, leading to a lower resolution output with decreased contrast. The distinction between max and average pooling is illustrated in Figure 10 with an example. In summary, the use of pooling layers in a CNN enables the preservation of

[4] https://medium.com/@sdoshi579/convolutional-neural-network-learn-and-apply-3dac9acfe2b6/

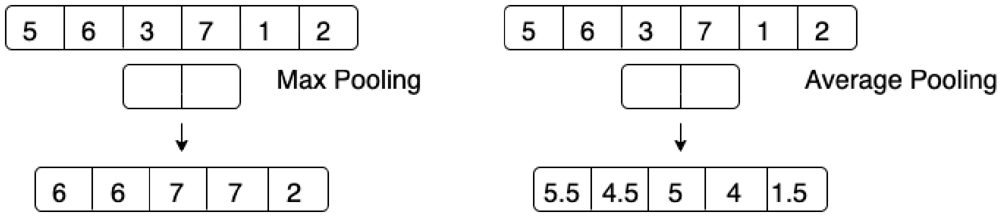


Figure 10: Comparison between the average and max pooling layers
(Assume the 1D pooling size is 2, the stride is 1, and padding is not considered)

spatial information in the input data while simultaneously decreasing dimensionality and computational demands (Szegedy et al., 2015). In the final step, the fully connected layer is employed to convert the output of pooling to the desired output size.

1.2. Literature Review

The literature on the subject of electricity price analysis is commonly classified into five areas: multi-agent models, fundamental methods, reduced-form models, statistical models, and ML and DL methods (Weron, 2014).

Multi-agent models typically focus on qualitative outcomes rather than quantitative ones. The Nash-Cournot framework (Borenstein et al., 1999), the supply function equilibrium (Baldick et al., 2004), strategic production-cost models (Batlle and Barquín, 2005), and agent-based simulation models (Guerci et al., 2010) are among the main sub-models within the multi-agent models category. These models, which are based on game theory, can provide insight into whether prices will exceed marginal costs and how this may impact the players' outcomes. However, they may encounter difficulties when more precise quantitative conclusions are required, such as when power price forecasts need to be made with a high degree of accuracy, stated in Weron (2014).

The fundamental method takes into account physical and economic factors that have an effect on electricity prices, e.g., demand and supply and weather conditions, and models their impact on prices based on mathematical equations. Two types of fundamental models are parameter-rich model (Johnsen, 2001) and parsimonious structural models (Eydeland and Wolyniec, 2002). The primary benefit of the method discussed in Weron (2014) is its high interpretability. However, it is important to note that there are two significant challenges that may arise from this method. Firstly, these models are primarily designed for medium-term forecasting rather than short-term forecasting. Secondly, the models perform optimally under specific physical and economic conditions.

Reduced-form models, as discussed in Weron (2014), offer a simplified and salient representation of key features of daily electricity prices, i.e., marginal distributions and price fluctuations. These models can be broadly categorized into two main areas: Jump-diffusion models, as outlined in Albanese et al. (2012), and Markov regime-switching models, as discussed in Janczura and Weron (2010). Like fundamental models, these models are not intended for forecasting short-term prices and are more suitable for medium-term forecasting.

Statistical and ML models have captured researchers' attention in recent years because they are able to make short-term forecasts. Moreover, statistical and ML models have been replaced with fundamental, reduced-form, and multi-agent models in recent research. According to Weron (2014), a significant proportion of publications in the field of EPF utilize either time series models or neural network models because statistical and machine learning methods have been demonstrated to produce the most favorable results.

Specifically for the class of time series and ML models, the class can be categorized into univariate and multivariate forecasting settings. According to Gürtler and Paulsen (2017), in the context of statistical time-series modelling, the most widely employed models in a univariate setting are the AR, the Auto-Regressive Moving Average (ARMA), and the ARIMA model. An ARMA is the ARIMA model without the integrated part, and it models the actual values rather than differences. Also, as seen in numerous pieces of electricity price forecasting research, AR and ARIMA were employed as baseline models in the model comparison (Xie et al., 2013; Weron and Misiorek, 2008; Peng et al., 2018). Along with considering univariate time-series models in various electricity markets, e.g., AR in Lebanon (Saab et al., 2001), ARMA in New England (Liu and Shi, 2013) and in Australian and PJM markets (Yang et al., 2017), and ARIMA in Spanish and Californian markets (Contreras et al., 2003; Zhou et al., 2004), the ARIMA model was used in the Ontario electricity market either on its own (Zareipour, 2012) or hybridized with other models (Conejo et al., 2005).

Regarding time-series models in the multivariate setting, the univariate time-series models were developed, and exogenous variables were incorporated into the models to improve the forecasting result. Among them, the vector autoregressive model (VAR) captured the EPF researcher's attention, e.g., Haldrup et al. (2010) in the Nord Pool grid, Ziel and Weron (2018) in European electricity markets, and Paschen (2016) in the German electricity market, showing the superiority of the VAR model compared with AR.

A limitation of statistical models is that they tend to be linear forecasters and they may not perform well with high-frequency data, i.e., hourly electricity price data with fluctuations (Lago et al., 2018). Statistical models tend to perform well when data frequency is low, i.e., monthly and weekly data. However, when the data frequency is high, e.g., hourly prices, the complexity of nonlinear behavior becomes difficult to predict (Amjady and Hemmati, 2006). To predict the nonlinear relationship of price data, various ML models have been proposed. For example, KNN, DT, RF, SVR, and ANN are widely used for electricity price forecasting. In terms of the application of DT in EPF, Fragkioudaki et al. (2015) concluded that the DT model captured price spikes and predicted day-ahead electricity prices in European electricity markets. RF, known as an ensemble DT, was compared with the ARMA model by Mei et al. (2014) in the NYISO and RF improved error metrics.

Additionally, Sansom et al. (2003) found that the SVR model requires less time to train than ANN and performs a more accurate forecast in the New South Wales market. Also, in the Ontario electricity market, the study conducted by Intan Azmira et al. (2020) found that a developed SVM model called the least square support vector machine (LSSVM) outperformed ANN, DT, and ARIMA models. On the other hand, by increasing the number of hidden layers of ANN, electricity price forecasting can be drastically improved compared with SVM, ARIMA, and VAR (Panapakidis and Dagoumas, 2016). The same result was obtained in Pourdaryaei et al. (2019), representing the superiority of ANN compared to SVM in the Ontario market. They also enhanced the model performance using optimization techniques for hyperparameter tuning. To deal with the problem of time series forecasting, some structures of ANN models were extended and tailored for time series forecasting to capture patterns by considering data as a sequence.

DL models have been shown to be highly effective in addressing sequence modelling problems, e.g., computer vision, audio signal processing, and natural language processing (Zhang et al., 2019). DL model is an ANN model, consisting of multiple recurrent layers of neurons capable of learning sequential data. While data is considered in a sequence in which one data point is reliant on the previous data point, the structure of ANN should be modified to learn the dependencies between data points. For modelling sequential data, RNN is the most widely used method. RNNs utilize the concept of memory to store the information from previous inputs, which is then used to generate the next output of the sequence (Schuster and Paliwal, 1997).

Although the RNN model outperforms ARIMA for modelling unstable sequences with large volatility (Jetcheva et al., 2014), gradient vanishing and explosion problems can easily occur because it employs recursion to extract the information from a sequence and it forgets the information as the sequence length increases (Pascanu et al., 2013). To cope with the issue of missing important information in long sequences, GRU and LSTM were proposed (Chung et al., 2014; Hochreiter and Schmidhuber, 1997). Hence, these models can help solve the gradient vanishing and explosion problems of standard RNNs. In the comparison between GRU and LSTM models, GRU converges faster because of a lower number of parameters, but both tend to reach the same level of accuracy (Rahman et al., 2018; Wang et al., 2021).

RNN-based models, including RNN, LSTM, and GRU, are widely used in electricity price forecasting. In a comparison between GRU and LSTM for a day-ahead electricity forecast in the Turkish market, it was found that GRU slightly reduced the error compared to LSTM (Ugurlu et al., 2018). Also, in the European electricity market, the GRU model outperformed the LSTM (Lago et al., 2018). However, LSTM hybridized with an attention mechanism outperformed GRUs in the Danish market (Meng et al., 2022). Additionally, both LSTM and GRU provide better results compared with SL and ML time series models (Lago et al., 2018; Peng et al., 2018).

CNN offers an alternative to RNN-based models for modelling sequential data. CNN extracts features through a pair of convolution and pooling operations (Sezer et al., 2020). CNN is widely utilized for feature extraction in recent price and load forecasting research in electricity markets, combined with other methods. Notably, employing only a convolutional layer may not improve the efficacy of a model or outperform RNN-based models (Ugurlu et al., 2018; Son, 2021) but it is faster at converging. However, building the complex model structure based on CNN layers by stacking them and using dilated convolutional layers can enhance the forecasting result in the Ontario market (Deng et al., 2021).

Additionally, hybrid models are becoming increasingly prevalent. Among these hybrid models, architectures that incorporate both CNN and RNN models are particularly popular. CNN layers extract spatial features, while RNN layers extract temporal features from input data; thus, the hybrid model can capture both temporal and spatial patterns in sequences (Chung et al., 2022). There are two ways for the hybridization of CNN-based and RNN-based models, including serial and parallel. In a serial way, the CNN layers capture the price data at the beginning, then the CNN output is flattened to be input into the RNN model (Chung et al., 2022). The price is subsequently forecasted through fully connected dense layers. Most of the serial hybridization models were implemented in forecasting electricity load, e.g., CNN-LSTM (Guo et al., 2020; Rick and Berton, 2022), CNN-Bi LSTM (Khan et al., 2021), and only one study, based on CNN-GRU, proposed the method for the EPF problem. The serial hybridization of CNN and GRU layers in Yang and Schell (2022a) demonstrated the improvement in forecasting volatile price spikes in the New York electricity market compared to ARIMA, LR, CNN, and GRU.

A demerit of serial hybridization is that the extracted features from the structure of convolutional layers may affect the training of RNN layers. Then, the extraction of temporal features from spatial patterns may not be implemented properly. To solve the training problem, the parallel hybridization of the aforementioned models was proposed by Farsi et al., 2021 in electricity load forecasting. Although most of the research was carried out in electricity load forecasting (Li et al., 2022a; Chung et al., 2022), Lehna et al., 2022 employed the parallel CNN-LSTM model in EPF and proposed Naive-CNN-LSTM by averaging the result of CNN-LSTM and the naive results. The result showed a stark improvement compared with LSTM, VAR, and ARIMA in the German market.

Along with various forecasting models, two critical factors, including exogenous variables and target electricity markets, affect the forecasting result and make the comparison between studies complicated. Considering various electricity markets, prominent markets in the USA, e.g., New York, California, and New Jersey; Canada, e.g., Ontario and Alberta; and Europe, e.g., Spain, Germany, Nord Pool, and the UK, considered various pricing methodologies based on their national regulations (Gürtler and Paulsen, 2017). In terms of independent variables, together with the price of electricity, electricity demand, equal to electricity load, is the highly chosen variable in the literature (Gürtler and Paulsen, 2017). Other independent variables used in the EPF problem include the import and export of electricity (Fragkioudaki et al., 2015; Paschen, 2016), weather data, e.g., temperature, humidity, and precipitation (Lehna et al., 2022; Yang and Schell, 2022a; Haldrup et al., 2010), and calendrical data, e.g., hour, day, month, and season (Haldrup et al., 2010; Panapakidis and Dagoumas, 2016). Due to recent expansions of renewable energies, the impact of renewable energy sources on electricity prices was examined by considering the generation data from each type of electricity producer (Meng et al., 2022; Fragkioudaki et al., 2015). However, considering all variables in the EPF problem does not always improve the forecasting power, and it may reduce the performance of the model (Rodriguez and Anders, 2004), leading to only considering previous prices and proposing univariate forecasting models (Peng et al., 2018).

Table 1: Comparative study of research in the electricity market forecasting

Article	Model Type	Method	Market	Data	Horizon single	Horizon multi	Variables
Saab et al., 2001	S	AR	Lebanon	1 Mo	*		Price
Liu and Shi, 2013	S	ARMA-GARCH	New England	1 H		*	Price
Liu and Shi, 2013	S	ARMA	PJM, Spain, Australia	1 H	*		Price
Contreras et al., 2003	S	ARIMA	Spain, California	1 H		*	Price
Zhou et al., 2004	S	ARIMA	California	1 D	*		Price
Zareipour, 2012	S	ARIMA	Ontario	1 H		*	Price
Conejo et al., 2005	S	ARIMA-WAVELET	Spain	1 H	*		Price
Haldrup et al., 2010	S	VAR	Nord Pool	1 H	*		Price - Weather - Calendar
Ziel and Weron, 2018	S	VAR	European markets	1 H		*	previous Prices
Paschen, 2016	S	VAR	Germany	1 D		*	Generation - export - Price
Sansom et al., 2003	ML	SVR	New South Wales	30 M		*	previous Prices - Demand
Panapakidis and Dagoumas, 2016	ML	ANN	Italy	1 H	*		Price - Demand - weather - Calendar - Gas price
Pourdaryaei et al., 2019	ML	ANN	Ontario	1 H	*		Price - Demand
Intan Azmira et al., 2020	ML	LSSVM	Ontario	1 H	*		Price - Demand
Fragkioudaki et al., 2015	ML	DT	European markets	1 H	*		Generation - Import and Export - Price - Demand
Mei et al., 2014	ML	RF	New York	5 M	*		Price
Ugurlu et al., 2018	DL	GRU, LSTM, CNN	Turkey	1 H	*		previous Prices - Demand - Weather
Lago et al., 2018	DL	GRU, LSTM, CNN	France - Belgium	1 H	*		Price - Demand - Generation - Coal and Gas price
Meng et al., 2022	DL	LSTM-ATT	Denmark	1 H		*	Generation - Demand - Price
Peng et al., 2018	DL	LSTM	European markets	1 H	*	*	Price
Deng et al., 2021	DL	Deep CNN	Ontario	1 H		*	Price - Demand - Generation
Yang and Schell, 2022a	DL	GHTnet (GRU-CNN)	New York	5 M		*	Price - Volatility - Seasonality
Lehna et al., 2022	DL	CNN-LSTM	Germany	1 H		*	Price - Demand - Weather - Coal and Gas Prices
Our Study	DL	TriConvGRU (CNN-GRU)	Ontario	1 H	*	*	Price - Demand - Weather - Import and Export- Generation

* **Guide:** S: Statistical time-series, ML: Machine Learning, DL: Deep Learning, Mo: Month, D: Day, H: Hour, M: Minute

1.3. Contribution

Through the recent advancements in predictive models for the EPF problem, we tackle the EPF problem by providing a comprehensive analysis of time-series forecasting models, including statistical, ML, and DL models. As mentioned in the literature, DL models offer an acceptable result for forecasting problems in the electricity market. Now, the first research question is: Does ensembling DL models improve price forecasting in the Ontario electricity market, compared to basic DL, ML, and statistical models? To deal with the problem, a tri-head parallel CNN-GRU (TriConvGRU-TCG) with various input time-series data is proposed. The raw data is input into the model concurrently through three streams. The first stream processes the raw data directly using a parallel CNN-GRU, while the other two streams first apply average pooling layers to downsample the input data. Therefore, each stream, consisting of a parallel CNN-GRU, receives input data with different frequencies. Also, the second research question is whether incorporating independent variables into our model is effective on the forecasting power or not. To answer the question, all baseline models as well as the proposed model are assessed in two settings: multivariate and univariate settings. The contributions of this paper can be summarized as follows:

1. The TriConvGRU (TCG) model is proposed to forecast short-term electricity prices in Ontario electricity market. The proposed model uses the structure of both parallel and consecutive hybrid models, based on CNN and GRU.
2. To verify the effectiveness of the proposed model, it is compared with statistical models, including *AR*, *ARIMA*, and *VAR*, ML models, including *LR*, *SVR*, *KNN*, and *DT*, and DL models, including *CNN*, *LSTM*, *GRU*, *Parallel CNN-GRU*, and *consecutive CNN-GRU*. Note that our study assesses the efficiency of baseline models in the Ontario electricity market, used to forecast electricity markets, along with our proposed model. The performance of various models was evaluated by calculating Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) metrics. The results show that the forecasting performance of the proposed model for electricity price forecasting tasks is

greatly improved.

3. Aside from the models mentioned before, the result of the proposed model is compared with the forecasted prices provided by IESO. The comparative result illustrates that the model can easily outperform multi-step ahead projections of the IESO company.
4. In our study, the proposed model as well as other forecasting models are evaluated in univariate and multivariate settings. Exogenous variables, including demand, import and export, generation, and meteorological data, are gathered with electricity prices for comparison. The final comparisons demonstrate the superiority of the univariate setting over the multivariable condition, particularly for the proposed model.

The remainder of the paper is structured as follows. Section 2 introduces mathematical notations and describes the proposed model, along with the datasets used. Section 3 assesses the capability of our model to forecast the IESO electricity market price and compares the model with baseline models. Finally, Section 4 summarizes the results and explains potential ways for future research.

2. Methodology

This section presents the dataset and its descriptive statistics, followed by an explanation of the preprocessing steps. Next, the proposed forecasting model is explained and visualized. Lastly, the optimal parameters for the model are determined, and the evaluation metrics used in the study are outlined.

2.1. Dataset

In this analysis, we choose the hourly Ontario electricity price as our target variable, considering the hourly price series from 1^{st} of January 2021 to 31^{st} of June 2022 for analysis purposes, as shown in Figure 11. It is important to note that the market operates 24 hours a day and 7 days a week, which contributes to the continuous nature of the data. The principle objective of this work is to forecast one-hour, two-hour, and three-hour ahead electricity prices for three separate weeks in 2022, considering the hourly price from the previous year (2021). Note that in our research, the word 'step' is used interchangeably with the word 'hour'. For this purpose, the training, validation, and test set data are required to make the model effectively workable. Therefore, three weeks in the first six months of 2022 were selected randomly as the test set required to assess the model in time periods. In order to find the optimal values for hyperparameters and avoid overfitting problems in training our models, the last 20% of the dataset, corresponding to 2021, is allocated to the validation set. The different forms of datasets with statistical analysis for the target variable and how they are used in the forecasting process are shown in Table 2.

In addition to the hourly Ontario electricity price (HOEP), several exogenous variables were selected to determine whether these variables enhance the forecasting performance or not. The summary of variables is in Table 3. In terms of independent variables, the electricity demand, the import and export of electricity, weather data, as well as generation data, were taken into account after reviewing studies in the literature. To consider the electricity demand, we decided to include the market demand and Ontario electricity demand variables.

Table 2: Statistical analysis of electricity prices in the datasets

Summary of Statistical Analysis of Electricity Prices						
Data set	**Time Period**	**Mean**	**Min**	**Max**	**Std**	**Count**
Train set	Jan 1st 2021 - Oct 20th 2021	24.34	-3.94	1660.80	33.27	7066
Validation set	Oct 21st 2021 - Dec 31st 2021	35.78	-0.11	190.88	17.76	1766
Test set-week 1	Jan 27th 2022 - Feb 2nd 2022	42.48	0	73.84	16.81	168
Test set-week 2	Feb 28th 2022 - Mar 7th 2022	34.20	0	68.11	17.24	168
Test set-week 3	May 12th 2022 - May 19th 2022	52.49	0	108.29	28.49	168

* **Note: The unit of prices are** $\$/MWh$

Table 3: Statistical analysis of variables

Summary of Statistical Analysis of Variables						
Type	**Features**	**Mean**	**Std**	**Min**	**Max**	**Source**
Price	Electricity price ($/MWh)	29.53	31.90	-4.43	1660.8	IESO data directory [5]
Meteorological	Dew point (°C)	1.47	11.21	-29.85	23.1	Canada Weather Stats [6]
	Temperature (°C)	7.32	11.79	-23.8	32.7	
	Relative humidity (%)	69.34	17.10	16.5	100	
Demand / Import and Export	Ontario demand (MW)	15408.49	2294.69	10426	22906	IESO data directory [5]
	Market demand (MW)	15479.86	2294.15	10595	22909	
Generation	Nuclear (MW)	9328.30	884.88	5893	10864	IESO data directory [5]
	Gas (MW)	1500.79	1465.68	68	7107	
	Hydro (MW)	4024.39	775.24	2269	6345	
	Wind (MW)	1482.92	1113.05	6	4586	
	Solar (MW)	79.76	117.95	0	433	
	Biofuel (MW)	35.38	33.35	0	233	

Note that the effect of exported amounts of electricity is considered in the Ontario demand and the imported amount is considered in the market demand, as explained in Section 1.1.1. As the second group of variables, the effect of various kinds of electricity generators on price, including hydroelectric, nuclear, bio-fuel, solar, wind, and gas producers, was considered based on the amount of electricity produced by each plant. By including the variables, the direct impact of renewable electricity production on the hourly price data is considered in our study. Aside from the demand and generation data, the previous price of electricity is included in the variables to better capture the trends. All data corresponding to electricity demand, price, and generation are gathered from the IESO data directory.[5] Finally, in the case of meteorological data, the dew point, temperature, and relative humidity are gathered from Canada Weather Stats.[6]

[5] https://www.ieso.ca/en/Power-Data/Data-Directory
[6] https://www.weatherstats.ca/

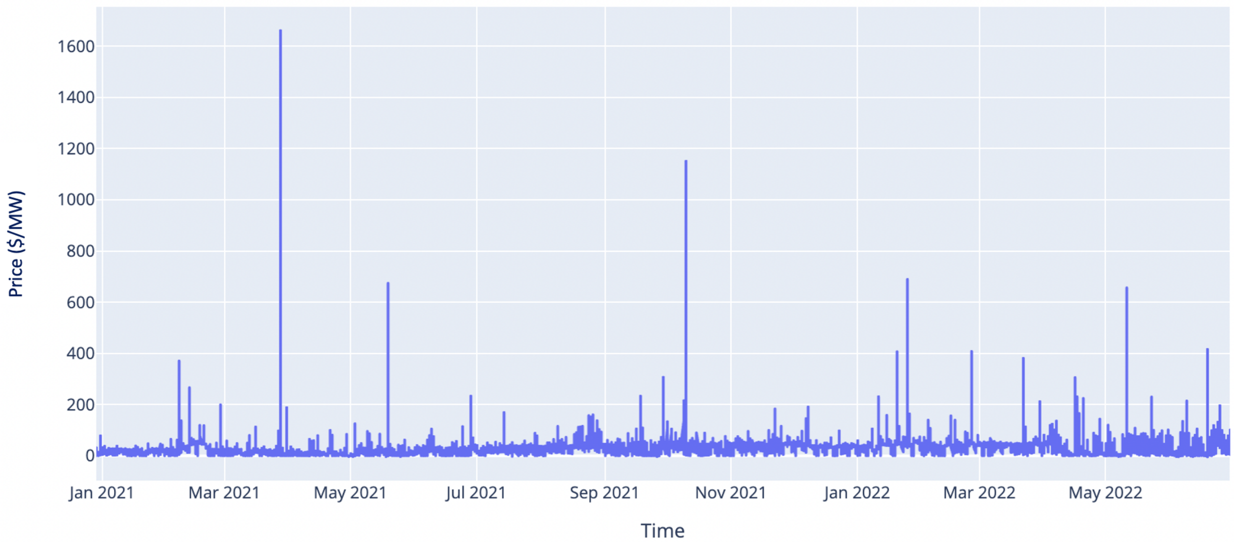


Figure 11: Hourly electricity price over time in Ontario Province from January 2021 until June 2022

2.2. Data Preprocessing

The first data preprocessing step in our study is to find the most relevant and important variables from the dataset, described in Section 2.1, for the reduction of the modelling complexity of time-series forecasting. Therefore, the complexity of our problem can be reduced by employing feature selection techniques. The Pearson correlation coefficient (PCC) is a classical method for measuring the linear correlation between two variables (Niu et al., 2022). The value of PCC ranges from $+1$ to -1, with a greater absolute value indicating a stronger correlation. The correlation between variables X and Y is represented as $PCC_{X,Y}$ and calculated as follows:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E\left[(X-\mu_X)(Y-\mu_Y)\right]}{\sqrt{D_X}\sqrt{D_Y}} \tag{12}$$

where $cov(X,Y)$ is the covariance of the variables X and Y; σ_X and σ_Y are the standard deviations of X and Y. The mean of X and Y are represented by μ_X and μ_Y, and the variances of X and Y are represented by D_X and D_Y, respectively.

Table 4: Interpretation of a size of the PCC (Mukaka, 2012)

Correlation size	**Explanation**
0.9 – 1 or -0.9 – -1	Very strong positive/negative correlation
0.7 – 0.9 or -0.7 – -0.9	Strong positive/negative correlation
0.5 – 0.7 or -0.5 – -0.7	Moderate positive/negative correlation
0.3 – 0.5 or -0.3 – -0.5	Weak positive/negative correlation
0 – 0.3 or 0 – -0.3	Very weak positive/negative

The strength of the PCC correlation can be classified into different tiers, from very weak to very strong (Mukaka, 2012; Akoglu, 2018), as shown in Table 4. As observed in Figure 12, the PCC calculation result reveals that the correlation between the electricity price and features is not significantly strong because all variables are less than 0.5, which is the threshold for a strong correlation. However, in this study, variables with correlations greater than 0.2 are considered features in our analysis because boundaries are not strict and may vary a bit, involving *Ontario demand*, *market demand*, *Hydro*, *Gas*, and $price_{t-1}$. Moreover, the meteorological data, as well as *Nuclear*, *Solar*, and *Biofuel* are eliminated because they have a very weak correlation with the target variable. Therefore, our study considered both multivariate and univariate forecasting to determine whether incorporating variables improved the forecasts or not.

In the second step, to handle inputs of varying magnitudes and dimensions, the data must be normalized. The normalization in our study was conducted column-wise, which means that data per each feature is normalized. The input data was normalized using Eq. 13:

$$x_n^k = \frac{x_o^k - x_{min}^k}{x_{max}^k - x_{min}^k} \tag{13}$$

where x_n^k represents the normalized data in kth feature, x_o^k represents the original datum in kth feature, x_{max}^k represents the maximum datum in kth feature, and x_{min}^k represents the minimum datum in kth feature.

In the next step, to process time series using DL models, it is necessary to create time windows during the preprocessing stage (Hossain et al., 2021b). Figure 13 illustrates how

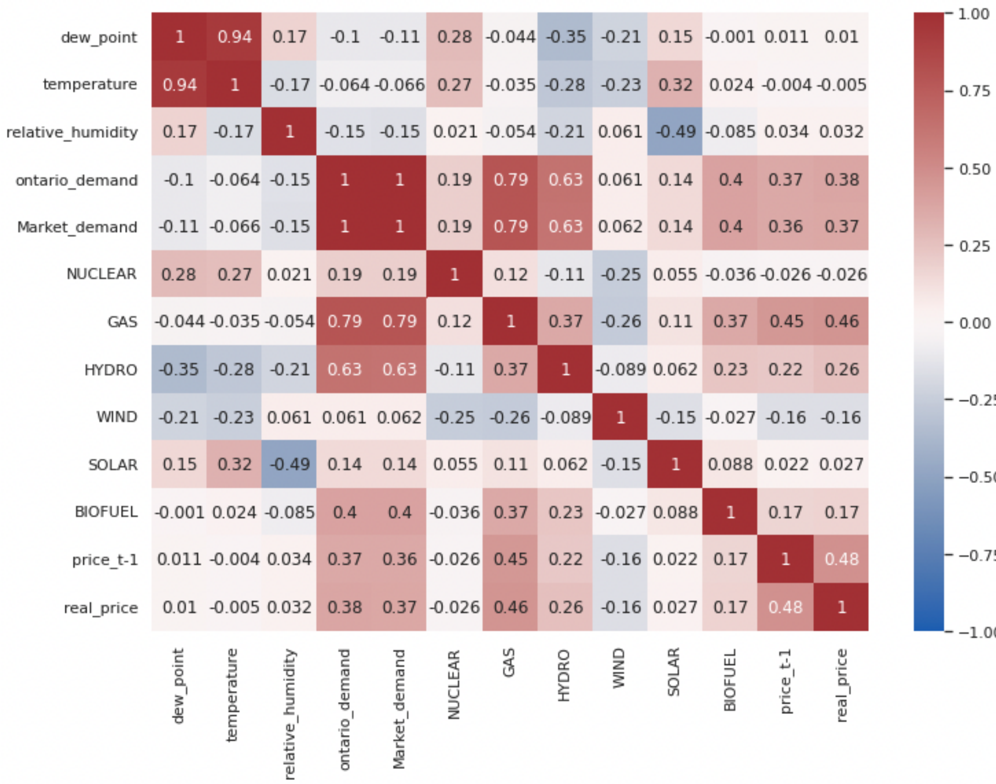


Figure 12: PCC between the Ontario electricity price and different features

the sliding window approach is then employed to construct the inputs and outputs of the DL models. In this approach, the window size is represented by W, the forecasting horizon is represented by H, the input sequence is represented by X, and the output data at each time step is represented by y. The time-series data is partitioned into windows of size W, with each window serving as input and the data within the forecasting horizon serving as output. The time-series data is divided into chunks of size W, with each chunk serving as input and the data within the forecasting horizon serving as output. The window is then advanced by one step and the process is repeated to generate time sequences for each time step.

As it is shown in Figure 14a, the way for building time sequences/windows for univariate time-series forecasting is presented. Each time window includes data corresponding to the previous T time periods for predicting the desired horizon, considered the next three hours in our study. In other words, the model uses price data from time zero to time T to indicate

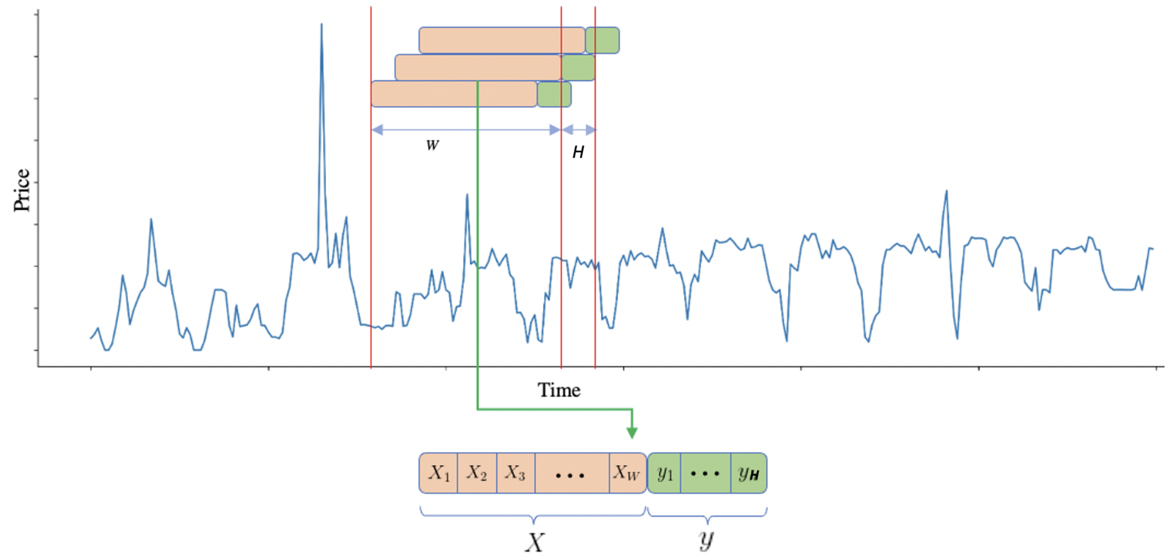


Figure 13: The sliding window approach for data preparation (Hossain et al., 2021b)

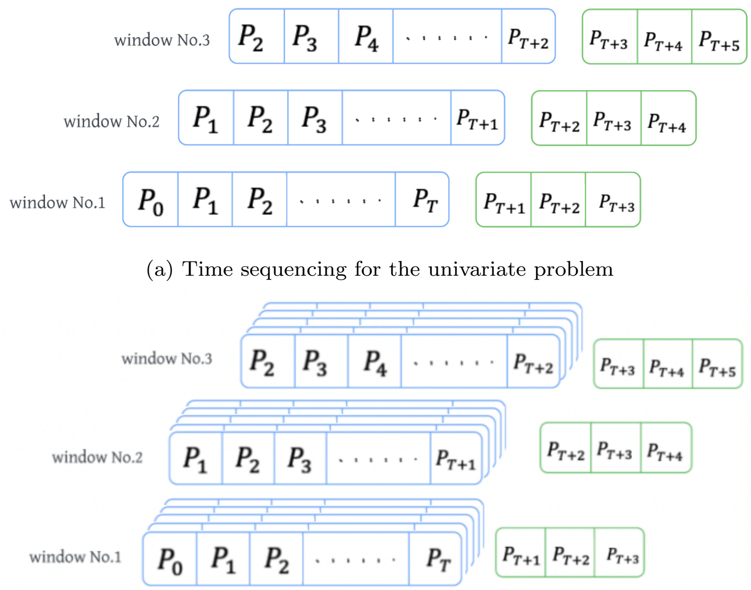
the electricity price for time $T+1$, $T+2$, and $T+3$. On the other hand, the sequencing for multivariate time-series forecasting is quite different due to exogenous variables, as shown in Figure 14b. In the multivariate problem, the other four variables along with the previous price data are incorporated for the input, and the output remains untouched.

2.3. Proposed Model

The proposed model is inspired by the ensemble of RNNs and CNNs models mentioned in Section 1.1.2 which are designed to solve the EPF problem. The proposed model comprises two major parts, as depicted in Figure 15:

- A convolution section, including 1D CNN layers with 1D kernels, 1D pooling layers, and the ReLU activation function which are illustrated in orange, red, yellow, and green, respectively, to extract important spatial features from the input price data
- A recurrent section with GRU units, situated after the convolution section, to encode the sequence and extract the temporal features which are illustrated in purple

In the first step, the preprocessed data is entered into each stream of the model concurrently. In the first stream (T), the raw data is modelled directly by a CNN but in the other two streams ($\frac{T}{4}$ and $\frac{T}{8}$), the input data (P) are first downsampled by average pooling layers



(a) Time sequencing for the univariate problem

(b) Time sequencing for the multivariate problem

Figure 14: The illustration of time sequences

with different kernel/filter sizes of 4 and 8, resulting into generating two input series (P' and P''). The pooling process is depicted in Figure 10. Thus, three sets of input sequential data of various lengths are generated, as described in Eqs 14a – 14c. This method allows the convolution layer to have multiple frequencies of price patterns during the learning process.

$$P = \{p_1, p_2, p_3, \cdots, p_T\} \tag{14a}$$

$$P' = \{p'_1, p'_2, p'_3, \cdots, p'_{T/4}\} = \left\{\frac{p_1 + p_2 + p_3 + p_4}{4}, \cdots\right\} \tag{14b}$$

$$P'' = \{p''_1, p''_2, p''_3, \cdots, p''_{T/8}\} = \left\{\frac{p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8}{8}, \cdots\right\} \tag{14c}$$

where P, P', and P'' are downsampled input sequences and $P \in R^{T\times v}$; R donates all time-series input sequences generated in the preprocessing phase; T refers to the initial input sequence length; and v refers to the number of variables used in the sequence. After

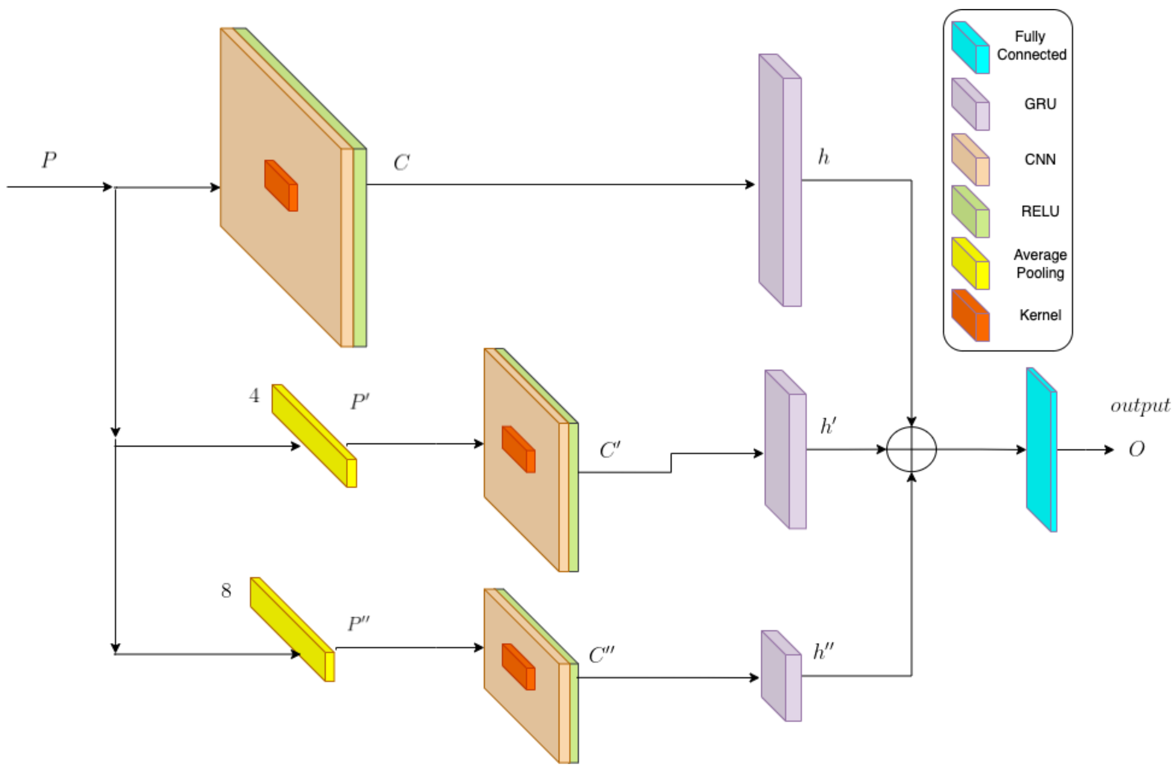


Figure 15: Illustration of the proposed model architecture
P shows the raw price data. First, we downsample the raw price data to generate P' and P'' as inputs for the second and third streams. The downsampled data, along with raw data, are fed into the convolution part. Outputs of convolution parts (C, C', and C'') are then entered into the recurrent part. Afterwards, the results of the three recurrent parts (h, h', and h'') are concatenated. Finally, the concatenated result passes through the fully connected layer and the output (O) is obtained.

downsampling, three data streams are fed into convolution layers with the ReLU activation function. To reduce the number of parameters for the convolution section, a downsampling technique is employed to preprocess the input data rather than using a convolution with a wider receptive field and stride. Following Eq 11, the output of the CNN part of each stream is as follows:

$$C = ReLu\left(W * P + b\right) \tag{15a}$$

$$C' = ReLu\left(W' * P' + b'\right) \tag{15b}$$

$$C'' = ReLu\left(W'' * P'' + b''\right) \tag{15c}$$

where C, C', and C'' refer to the outputs of convolution layers with different lengths that capture multiple frequencies of patterns; W and b are the weights and biases of the CNN layer, respectively.

In the second step, each convolution output is fed to a separate GRU network to encode the important features. The hidden state of GRU is formulated by Eq 10, mentioned in Eqs 16a – 16c.

$$h_t = GRU\left(h_{t-1}, c_t\right) \tag{16a}$$

$$h'_t = GRU\left(h'_{t-1}, c'_t\right) \tag{16b}$$

$$h''_t = GRU\left(h''_{t-1}, c''_t\right) \tag{16c}$$

where h_t, h'_t, and h''_t refer to the hidden states of GRU layers at each time step.

At the final step, the last hidden states of GRU outputs are concatenated at time t and the concatenated output passes through a fully-connected linear model to obtain the desired output shape, which is our forecasting horizon (o_t). The dimension of tensors at each step is calculated in Table 5.

$$o_t = W_t\left[h_t, h'_t, h''_t\right] + b_t \tag{17}$$

2.4. Baseline Models

To thoroughly evaluate the proposed model, a set of baseline models were constructed and a comprehensive analysis of the performance of each model was carried out for comparison. The baseline models include statistical time series models, such as AR, ARIMA, and VAR;

Table 5: Dimensions of tensors in each step

Step	Layer	Name	Dimension
0	Input data	P	[Batch_size, Seq_length, N_features]
0	Pooling layer (4)	P'	[Batch_size, Seq_length/4, N_features]
0	Pooling layer (8)	P''	[Batch_size, Seq_length/8, N_features]
1	CNN output (stream 1)	C	[Batch_size, Seq_length, Hidden_size_CNN1]
1	CNN output (stream 2)	C'	[Batch_size, Seq_length/4, Hidden_size_CNN2]
1	CNN output (stream 3)	C''	[Batch_size, Seq_length/8, Hidden_size_CNN3]
2	GRU hidden (stream 1)	h	[Num_layer, Batch_size, Hidden_size_GRU1]
2	GRU hidden (stream 2)	h'	[Num_layer, Batch_size, Hidden_size_GRU2]
2	GRU hidden (stream 3)	h''	[Num_layer, Batch_size, Hidden_size_GRU3]
3	Concatenation	-	[Batch_size, Hidden_size_GRU1+ Hidden_size_GRU2+ Hidden_size_GRU3]
3	Output	O	[Batch_size, Horizon_size]

* **Guide:** Batch_size: the size of batches, Seq_length: the length of a sequence, N_features: number of features, Hidden_size: the hidden size of layers (CNN or GRU), Num_layer: the number of GRU Layers, and Horizon_size: the forecasting horizon

ML models, such as SVR, KNN, LR, and DT; and DL models, such as CNN, LSTM, GRU, and hybrid CNN-GRU (consecutive and parallel). These models are introduced in Section 1.1.2.

Regarding the hybrid models mentioned in Section 1.2, the convolutional layer is used in the consecutive CNN-GRU architecture to derive important features from the input data. The GRU layers are then employed to extract long-term dependencies. (Alhussein et al., 2020), as demonstrated in Figure 16.

In contrast, the parallel CNN-GRU architecture uses a dual pathway, where the input price data is first directed into two streams for processing. The GRU and CNN pathways independently extract features present in the data and prepare the input for the final forecast, as illustrated in Figure 17. Notably, the performance of all baseline models as well as the proposed model heavily relies on the hyper-parameter tuning, considered in Section 2.5.

2.5. Hyperparameter Optimization

The performance of statistical, ML, and DL models is greatly influenced by hyperparameters, thus it is of utmost importance to set them correctly. In this study, a grid search algorithm was used with the aforementioned models. The approach evaluates all possible combinations of values within the defined range, in order to identify the model with the

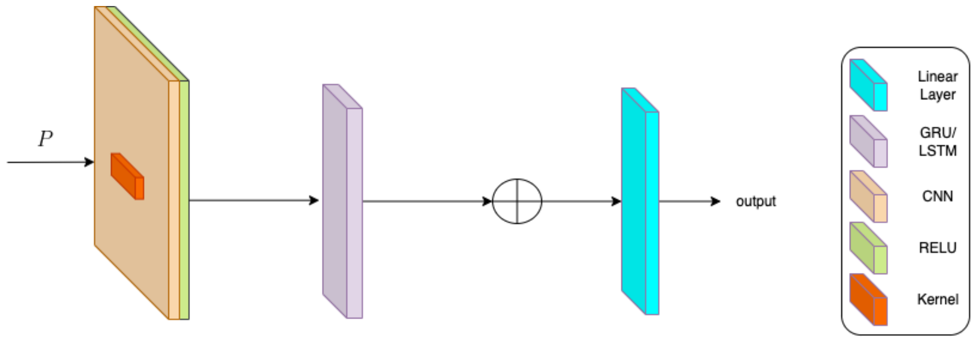


Figure 16: Representation of consecutive CNN-GRU model architecture

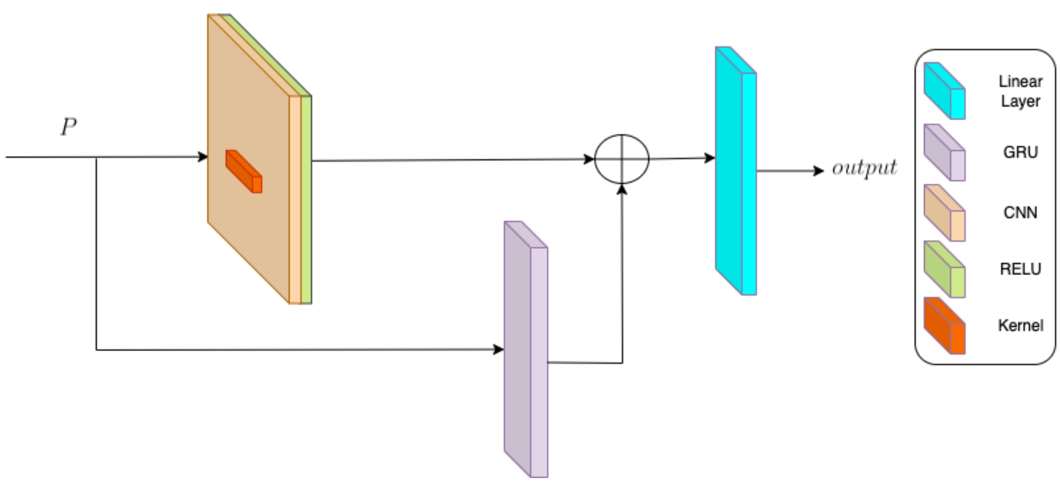


Figure 17: Representation of parallel CNN-GRU model architecture

Table 6: Summary of tuned hyperparameters for univariate forecasting models

Model	Optimal Hyperparameters
AR	$p = 25$
ARIMA	$p = 2,\ d = 1,\ q = 1$
SVR	$C = 0.1,\ epsilon = 0.01,\ kernel\_size = rbf$
DT	$max\_depth = 5$
KNN	$leaf\_size = 30, n\_neighbours = 20, p = 2, weight = distance$
LSTM	$LR = 0.001, hidden\_layer = 16, optimizer = Adam$
GRU	$LR = 0.001, hidden\_layer = 16, optimizer = Adam$
CNN	$LR = 0.001, hidden\_layer = 16, kernel\_size = 5, optimizer = Adam$
CNN-GRU(Parallel)	$LR = 0.001, hidden\_layer(CNN) = 32, hidden\_layer(GRU) = 64, Kernel\_size = 5, Optimizer = Adam$
CNN-GRU(Consecutive)	$LR = 0.0001, hidden\_layer(CNN) = 64, hidden\_layer(GRU) = 128, Kernel\_size = 9, Optimizer = Adam$
TriConvGRU(Proposed)	$LR = 0.001, hidden\_layer(CNN) = 128, hidden\_layer(GRU) = 128, Kernel\_size(head1) = 9,$ $Kernel\_size(head2) = 7,\ Kernel\_size(head3) = 5,\ Optimizer = Adam$

Table 7: Summary of tuned hyperparameters for multivariate forecasting models

Model	Optimal Hyperparameters
VAR	$p = 25$
SVR	$C = 0.1,\ epsilon = 0.02,\ kernel\_size = linear$
DT	$max\_depth = 50$
KNN	$leaf\_size = 30, n\_neighbours = 20, p = 2, weight = distance$
LSTM	$LR = 0.001, hidden\_layer = 128, optimizer = Adam$
GRU	$LR = 0.001, hidden\_layer = 32, optimizer = Adam$
CNN	$LR = 0.001, hidden\_layer = 16, kernel\_size = 9, optimizer = Adam$
CNN-GRU(Parallel)	$LR = 0.0001, hidden\_layer(CNN) = 128, hidden\_layer(GRU) = 128, Kernel\_size = 9, Optimizer = Adam$
CNN-GRU(Consecutive)	$LR = 0.001, hidden\_layer(CNN) = 16, hidden\_layer(GRU) = 16, Kernel\_size = 7, Optimizer = Adam$
TriConvGRU(Proposed)	$LR = 0.001, hidden\_layer(CNN) = 16, hidden\_layer(GRU) = 32, Kernel\_size(head1) = 9,$ $Kernel\_size(head2) = 7, Kernel\_size(head3) = 5, Optimizer = Adam$

lowest validation error. Note that the metric to calculate errors is a root mean square error (RMSE). The range for the hyperparameters used in the search was mentioned in Table A.15. Tables 6 and 7 show the optimization search results of multivariate and univariate forecasting for each model, respectively.

2.6. Evaluation Criteria

A systematic comparison of the forecasting performance of various models is conducted by utilizing multiple evaluation metrics. The most widely used metrics, root mean square error (RMSE) and mean absolute error (MAE), are implemented to demonstrate the forecasting accuracy of the different models. A summary of the definitions and formulas of these metrics is provided in Table 8. Furthermore, it is important to consider the stability of the models to ensure their resilience to external shocks in real-world applications (Zhang et al., 2022). The stability of the models is evaluated in this study by utilizing the variance of forecasting errors (var), as defined in Equation 18.

$$var = \frac{1}{T}\sum_{1}^{T}(e_t - \bar{e})^2 \tag{18}$$

where e_t shows the forecasting error at time t, and $\bar{e}$ shows the average of errors. Note that e_t is the squared difference between y_t and $\hat{y}_t$.

Also, the model confidence set (MCS) method, proposed by Hansen et al. (2011), is employed to evaluate the forecasting capabilities of various predictive models. By removing weaker models from the starting set, M^0, the MCS technique determines a superior set of models, $M^\star$. Two components comprise the MCS method: an equivalence test and an elimination rule (e_M). The equivalence test examines the null hypothesis, $H_{0,M}$, in contrast to the alternative hypothesis, $H_{A,M}$, at a specific significance level α.

$$\begin{aligned} H_{0,M} &: c_i j = 0 \text{ for all } i, j = 1, 2, \cdots, m \\ H_{A,M} &: c_i j \neq 0 \text{ for some } i, j = 1, 2, \cdots, m \end{aligned} \tag{19}$$

where $c_{i,j} = E(d_{i,j})$ is assumed be not time dependent; d_{ij} is defined as the difference between the losses of model i and model j, $d_{ij,t} = l_{i,t} - l_{j,t}$; $l_{i,t}$ is the loss function of model i at time t, $l_{i,t} = l(y_t, \hat{y}_{i,t})$.

If the null hypothesis, $H_{0,M}$, is not rejected, it leads to the termination of the MCS procedure; in this case, $M^\star_{1-\alpha} = M$. Conversely, if $H_{0,M}$ is rejected, the elimination rule is applied to remove a weak model from the model set before reinitiating the equivalence test process (Li et al., 2022b). Through the repetition of the procedure, the MCS test will ultimately yield a superior set of models ($M^\star_{1-\alpha}$) and a model with a greater p-value demonstrates superior forecasting accuracy.

Table 8: Summary of the evaluation metrics.

Evaluation metric	Description	Formula
RMSE	The square root of the average of the squared differences between T forecasted and actual values. (Weron, 2014)	$\sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(y_t - \hat{y}_t\right)^2}$
MAE	The average of the absolute errors between T forecasted and actual values.	$\frac{1}{T}\sum_{t=1}^{T}\lvert y_t - \hat{y}_t\rvert$

*** Note: T is the number of samples. y_t shows the real value at time t, and $\hat{y}_t$ represent the forecasted value at time t**

3. Empirical Results

In order to test the efficiency of the proposed model and compare it to the baseline models, we carried out a comparative analysis in three test sets representing three different weeks in 2022. We considered the comparison between univariate and multivariate settings, as well as between single-step ahead and multi-step ahead forecasting to evaluate the proposed model. Additionally, we performed the MCS statistical test to rank the forecasting models. Finally, we analyzed the stability of the proposed model, as well as the results of an ablation study, in our research.

3.1. Aggregated Univariate and Multivariate Forecasting

- The proposed model outperforms the IESO forecasts, which were used as the primary baseline, in both univariate and multivariate settings. As observed in Figure 18, the proposed model improved the IESO error noticeably by 63.7% in the first, 43% in the second, and 28.2% in the third week. It is worth noting that all DL models outperform the IESO baseline in univariate settings, and most DL models in multivariate settings do the same, with the exception of a few models in the third week. Our conclusion is that DL models, particularly in univariate settings, demonstrate significantly better results than the IESO forecasting model.

- We observe that when exogenous variables are used in DL models, results deteriorate except for some DL models in the first week. The result deterioration can be more visible in the third week when it has the most fluctuations compared to other weeks.

Although the result of the statistical model is not better than the proposed model, it is notable that adding independent variables helps improve the statistical models. As observed in Section 2.2, the previous price variable has the highest correlation with the current price, and other independent variables have low positive or negative correlations with the target. Thus, the inclusion of low-correlated features may lead to an overfitting problem in most cases for ML and DL models because the models perform poorly on test sets, particularly in the fluctuating test set.

- Based on the results shown in Table 9, our proposed model (TriConvGRU) in the univariate setting outperforms other models, including statistical, ML, and DL models. The improvement rate of our proposed model compared to other models in the univariate setting is represented in Figure 18. As observed in Table 10, GRU in the first week, LSTM in the second week, and LR in the third week outperform others in the multivariate setting. Although our proposed model in multivariate forecasting could not outperform others, the univariate TriConvGRU model still has the lowest forecasting error compared to all multivariate models. Therefore, we conclude that the univariate TriConvGRU outperforms all models in both settings.

3.2. One-step ahead Forecasting

The results of the one-step ahead comparison between the forecasted and actual values of baseline and proposed models are reported in the first rows of each model in Table 11 and 12. Important comparative points for the one-step ahead forecasting are listed below:

- Compared to the forecast made by IESO, the proposed model in the univariate setting reduced the RMSE error by 74.4% and the MAE error by 66.8% in the first week, 57.3% and 50.9% in the second week, and 33.2% and 26.1% in the third week. The sensible improvement can be observed in Figures 20a – 20c, which show the proposed model captures the ups and downs in test sets.

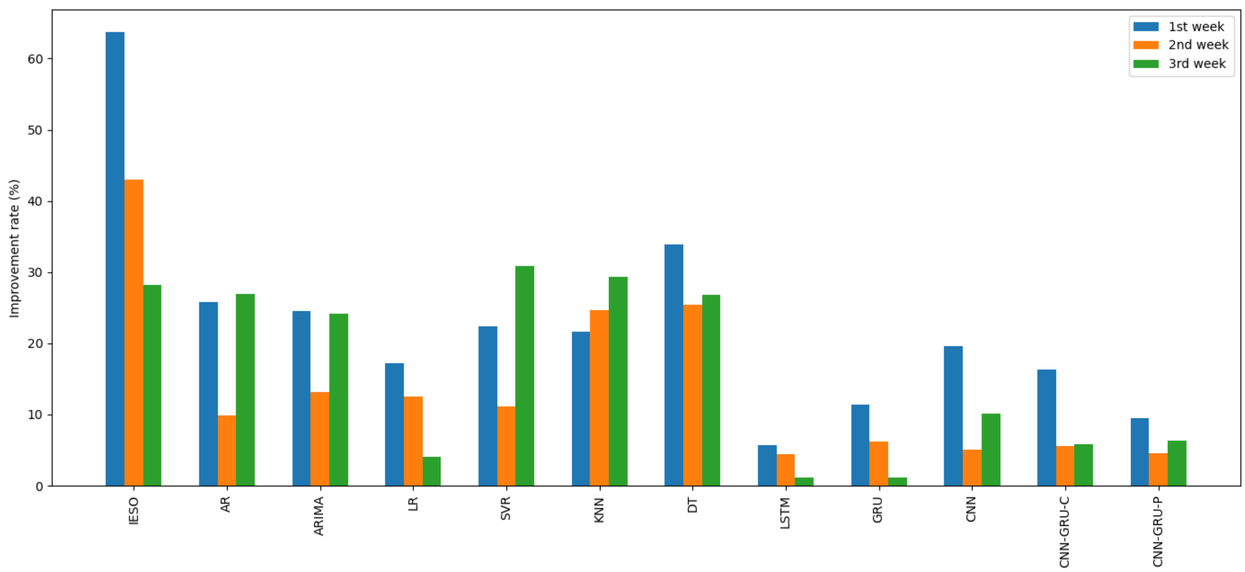


Figure 18: Improvement rates of the RMSE metric of the proposed model compared to the baselines in the univariate setting

Table 9: Average results of univariate forecasting models over forecasting horizons

Type	Model	27 Jan 2022 - 2 Feb 2022		28 Feb 2022 - 7 Mar 2022		12 May 2022 - 19 May 2022	
		RMSE	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
	IESO forecast	31.527	17.573	20.629	11.774	27.149	17.201
S	AR	15.423	13.815	13.067	11.141	26.675	24.303
	ARIMA	15.151	9.872	13.549	9.380	25.685	23.215
ML	LR	13.817	10.560	13.453	9.780	20.316	17.048
	SVR	14.739	12.134	13.236	10.169	28.228	23.688
	DT	14.593	10.656	15.603	10.839	27.557	22.613
	KNN	17.306	13.799	15.781	12.320	26.627	24.354
DL	LSTM	12.119	9.310	12.316	8.443	19.719	15.198
	GRU	12.908	9.483	12.550	8.723	19.706	15.117
	CNN	14.218	10.404	12.391	8.544	21.687	17.323
	CNN-GRU (Consecutive)	13.657	10.402	12.462	8.759	20.689	15.930
	CNN-GRU (parallel)	12.636	9.347	12.333	8.351	20.802	15.984
	TriConvGRU (Proposed)	**11.434**	**8.102**	**11.768**	**7.961**	**19.496**	**15.072**

⋆ Note: A univariate forecasting model refers to the use of only the price feature for forecasting. Also, Green and Bold numbers indicate the minimum RMSE error.

Table 10: Average results of multivariate forecasting models over forecasting horizons

Type	**Model**	**27 Jan 2022 - 2 Feb 2022**		**28 Feb 2022 - 7 Mar 2022**		**12 May 2022 - 19 May 2022**	
		RMSE	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
	IESO forecast	31.527	17.573	20.629	11.774	27.149	17.201
S	VAR	12.920	11.033	12.399	9.609	25.069	22.491
ML	LR	13.039	10.322	13.574	10.623	**22.103**	**19.505**
	SVR	14.629	11.636	19.358	16.355	24.982	22.611
	DT	21.527	13.482	21.010	12.450	51.908	34.698
	KNN	14.827	10.477	15.382	12.325	42.975	36.675
DL	LSTM	12.363	9.762	**12.360**	**9.418**	24.431	21.536
	GRU	**12.120**	**9.303**	12.531	9.590	23.938	20.921
	CNN	14.155	11.383	14.645	11.248	34.498	28.284
	CNN-GRU (Consecutive)	13.239	10.768	13.443	10.384	32.363	27.214
	CNN-GRU (parallel)	13.882	11.094	14.195	11.406	37.261	30.715
	TriConvGRU (Proposed)	12.465	9.705	12.985	9.797	25.860	21.458

*** Note: A multivariate forecasting model refers to the use of features mentioned in Section 2.2 for forecasting. Also, Green and Bold numbers indicate the minimum error.**

- In terms of one-step ahead forecasting comparison for ML and DL models, we saw that univariate forecasting is better than multivariate ones based on RMSE and MAE metrics in three sample weeks for each DL and ML model. Similar to the result of Section 3.1, we conclude that adding variables to ML and DL models deteriorates the one-step ahead results because it encountered the overfitting problem while the model complexity is increased.

- Regarding the comparison between DL models and ML and statistical models in the univariate setting, the improvement of results by DL models is discernible, as illustrated in Figure 19. Based on the RMSE metric, the best model for one-step ahead forecasting is the proposed model except for the third week. However, the proposed model outperforms others in all weeks based on the MAE metric. For multivariate one-step ahead forecasting, DL models follow a similar pattern. This indicates that DL models, particularly in the univariate setting, outperform ML and statistical models in one-step ahead forecasting.

- Although the proposed complex hybrid model improves the MSE or MAE metrics, the simple consecutive and parallel hybrid model did not outperform basic DL models,

Table 11: Experimental results of univariate forecasting models

Type	Model		27 Jan 2022 - 2 Feb 2022		28 Feb 2022 - 7 Mar 2022		12 May 2022 - 19 May 2022	
			RMSE	MAE	RMSE	MAE	RMSE	MAE
	IESO forecast	T+1	32.010	16.244	19.006	10.334	26.482	16.073
		T+2	33.690	19.1375	20.550	11.792	27.465	17.536
		T+3	28.881	17.338	22.331	13.196	27.501	17.995
S	AR	T+1	12.177	10.633	10.399	8.555	21.718	19.284
		T+2	16.002	14.340	13.564	11.583	27.600	25.157
		T+3	18.089	16.472	15.239	13.285	30.706	28.468
	ARIMA	T+1	12.332	8.037	10.971	7.550	21.448	18.958
		T+2	15.789	10.297	14.120	9.763	26.698	24.147
		T+3	17.331	11.282	15.556	10.826	28.910	26.541
ML	LR	T+1	11.688	8.849	11.106	8.069	18.100	14.989
		T+2	14.381	11.043	14.069	10.199	21.079	17.726
		T+3	15.382	11.789	15.183	11.071	21.770	18.430
	SVR	T+1	13.172	10.516	10.611	8.099	26.483	22.072
		T+2	14.808	12.192	13.537	10.371	28.475	23.911
		T+3	16.237	13.695	15.561	12.036	29.726	25.080
	DT	T+1	13.875	9.881	13.652	9.508	26.520	21.349
		T+2	14.406	10.471	15.942	11.036	27.597	23.412
		T+3	15.498	11.616	17.214	11.973	28.555	23.078
	KNN	T+1	15.788	12.758	14.294	11.180	25.743	23.488
		T+2	17.791	14.065	16.001	12.473	26.790	24.493
		T+3	18.339	14.573	17.049	13.307	27.347	25.082
DL	LSTM	T+1	9.089	6.142	8.518	5.538	**17.287**	12.223
		T+2	12.965	10.965	13.207	9.111	20.540	16.160
		T+3	14.303	10.822	15.224	10.679	21.329	17.210
	GRU	T+1	9.983	7.049	8.847	6.052	17.516	12.925
		T+2	13.893	10.293	13.601	9.460	20.378	**15.509**
		T+3	14.848	11.107	15.202	10.657	21.225	**16.918**
	CNN	T+1	11.430	8.147	8.744	5.729	18.625	14.249
		T+2	15.296	10.965	13.288	9.265	22.619	18.141
		T+3	15.924	12.101	15.141	10.638	23.816	19.579
	CNN-GRU (Consecutive)	T+1	10.076	7.227	8.802	5.784	17.957	12.770
		T+2	14.090	10.873	13.206	9.374	21.173	16.462
		T+3	16.804	13.106	15.378	11.120	22.936	18.559
	CNN-GRU (parallel)	T+1	9.760	6.581	8.357	5.340	18.518	12.938
		T+2	13.059	9.858	13.183	9.093	21.357	16.839
		T+3	15.089	11.601	15.459	10.619	22.532	18.175
	TriConvGRU (Proposed)	T+1	**8.177**	**5.393**	**8.106**	**5.070**	17.676	**11.879**
		T+2	**12.091**	**8.581**	**12.663**	**8.721**	**19.658**	15.559
		T+3	**14.033**	**10.333**	**14.534**	**10.093**	**21.153**	17.778

⋆ Note: $T+1$, $T+2$, and $T+3$ represent one-step, two-step, and three-step ahead forecasting, respectively. Green and Bold numbers indicate the minimum error.

Table 12: Experimental results of multivariate forecasting models

Type	Model		27 Jan 2022 - 2 Feb 2022		28 Feb 2022 - 7 Mar 2022		12 May 2022 - 19 May 2022	
			RMSE	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
	IESO forecast	T+1	32.010	16.244	19.006	10.334	26.482	16.073
		T+2	33.690	19.137	20.550	11.792	27.465	17.536
		T+3	28.881	17.338	22.331	13.196	27.501	17.995
S	VAR	T+1	10.036	8.470	9.535	7.562	20.436	17.950
		T+2	13.241	11.357	12.895	9.906	25.576	23.012
		T+3	15.482	13.271	14.768	11.360	29.195	26.512
ML	LR	T+1	11.969	9.498	11.297	8.905	**19.174**	**16.750**
		T+2	13.012	10.289	13.961	10.970	**22.938**	**20.172**
		T+3	14.136	11.179	15.465	11.993	**24.198**	**21.594**
	SVR	T+1	14.430	11.270	18.171	15.384	23.742	20.857
		T+2	14.353	11.511	19.729	16.804	25.294	23.079
		T+3	15.103	12.127	20.173	16.878	25.911	23.896
	DT	T+1	18.201	11.458	15.595	9.549	41.538	28.788
		T+2	22.800	14.340	21.982	12.825	60.810	38.633
		T+3	23.579	14.647	25.454	14.977	53.377	36.673
	KNN	T+1	12.993	10.360	14.952	11.921	42.416	36.175
		T+2	13.148	10.439	15.422	12.379	43.035	36.760
		T+3	18.339	10.632	15.773	12.674	43.475	37.090
DL	LSTM	T+1	10.187	7.839	**9.146**	**6.829**	22.909	19.683
		T+2	13.065	10.385	13.193	10.132	25.082	22.277
		T+3	13.838	11.062	14.742	11.293	25.303	22.649
	GRU	T+1	**10.082**	**7.481**	10.189	7.777	19.412	16.238
		T+2	12.849	**10.139**	**12.744**	**9.786**	25.023	22.045
		T+3	**13.430**	**10.288**	**14.661**	**11.206**	27.379	24.479
	CNN	T+1	12.862	10.304	12.763	10.311	37.161	30.345
		T+2	13.880	11.534	14.296	10.675	33.146	26.950
		T+3	15.722	12.310	16.875	12.759	33.188	27.557
	CNN-GRU (Consecutive)	T+1	12.078	9.956	11.688	9.125	27.762	23.098
		T+2	12.954	11.178	13.591	10.446	32.265	27.317
		T+3	14.686	11.170	15.051	11.581	37.062	31.226
	CNN-GRU (parallel)	T+1	13.330	11.129	12.390	10.145	30.149	23.638
		T+2	13.866	11.054	14.599	11.702	37.832	31.506
		T+3	14.451	11.100	15.597	12.371	43.801	37.001
	TriConvGRU (Proposed)	T+1	10.428	7.854	10.099	7.608	24.600	17.712
		T+2	**12.806**	10.147	13.122	9.817	26.431	23.317
		T+3	14.162	11.115	15.733	11.965	26.548	23.345

⋆ **Note: $T+1$, $T+2$, and $T+3$ represent one-step, two-step, three-step ahead forecasting, respectively. Green and Bold numbers indicate the minimum error.**

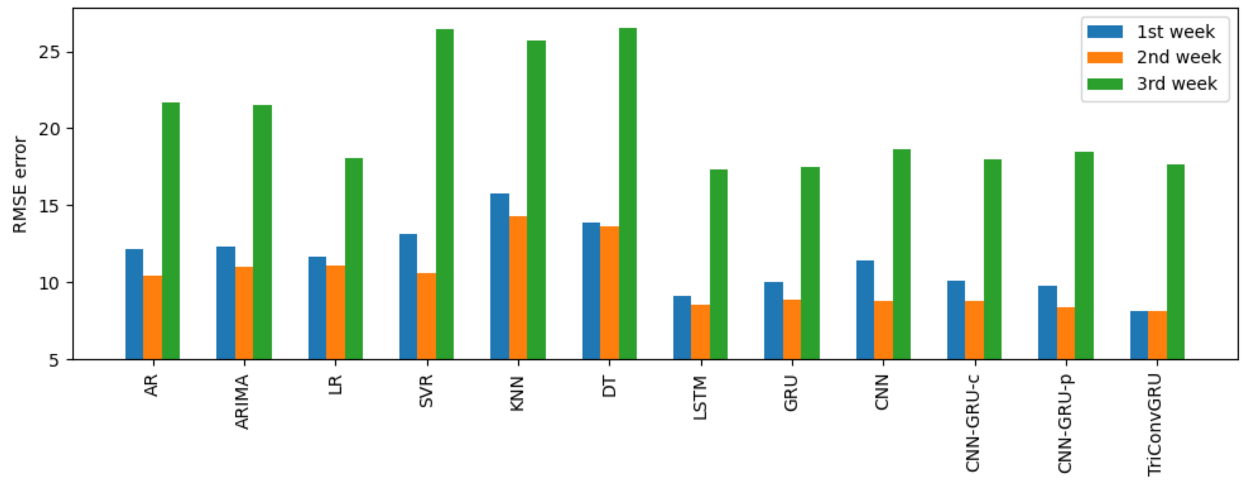


Figure 19: Comparison of RMSE metrics for univariate one-step forecasting

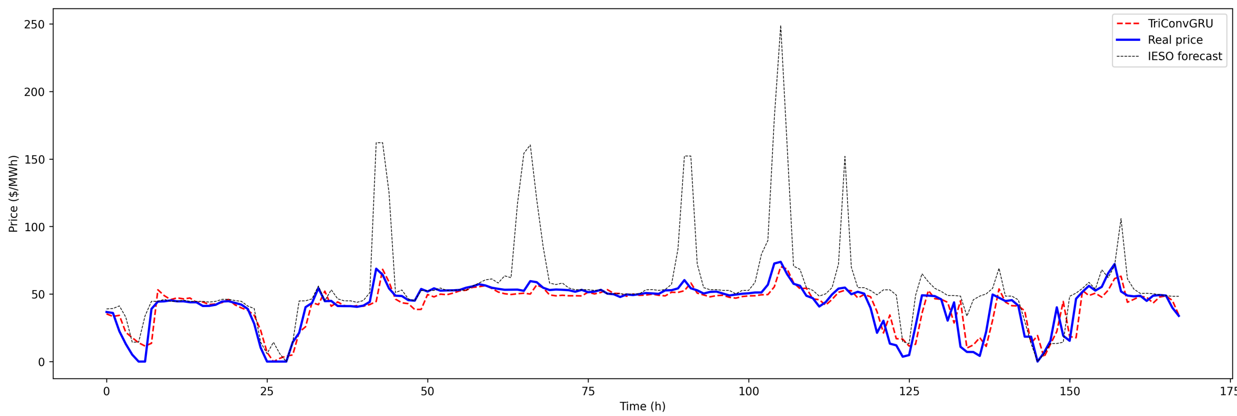
including LSTM and GRU. This finding implies that a hybrid model with more parameters is needed to improve the prediction result. So, we came up with a hybrid model with three different streams to capture various price patterns. Section 3.4 explains why our proposed model outperforms others.

- In the one-step ahead forecasting comparison between statistical models, the ARIMA model outperforms the AR model in a univariate setting. Also, the VAR results showed us that adding independent variables made the AR model better at making forecasts. Regarding ML models, LR outperforms other models in both univariate and multivariate settings.

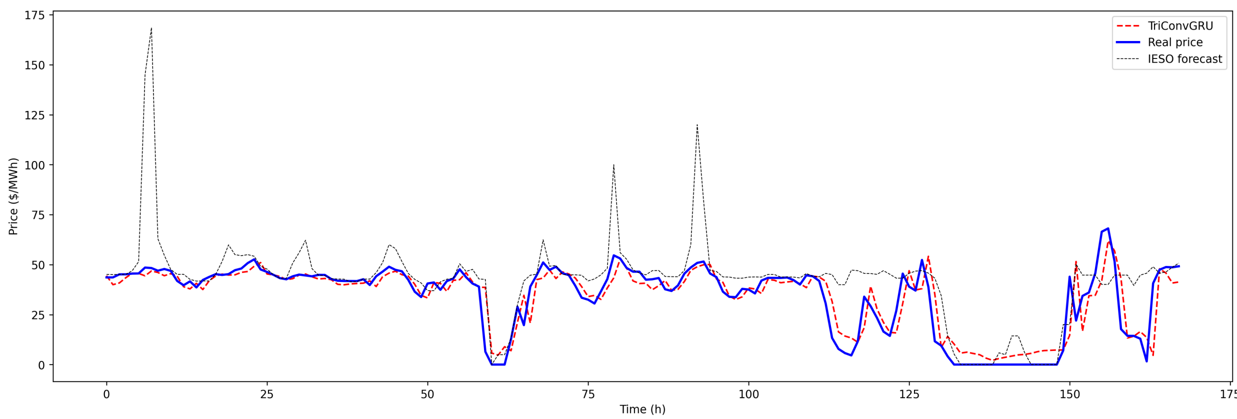
3.3. Multi-step ahead Forecasting

Along with one-step ahead forecasting, results for two-step and three-step ahead forecasting are presented in the second and third rows of each model in Tables 11 and 12, respectively. Obviously, the more the number of forecasting horizons, the more forecasting errors increase. Significant findings of comparison are listed in the following:

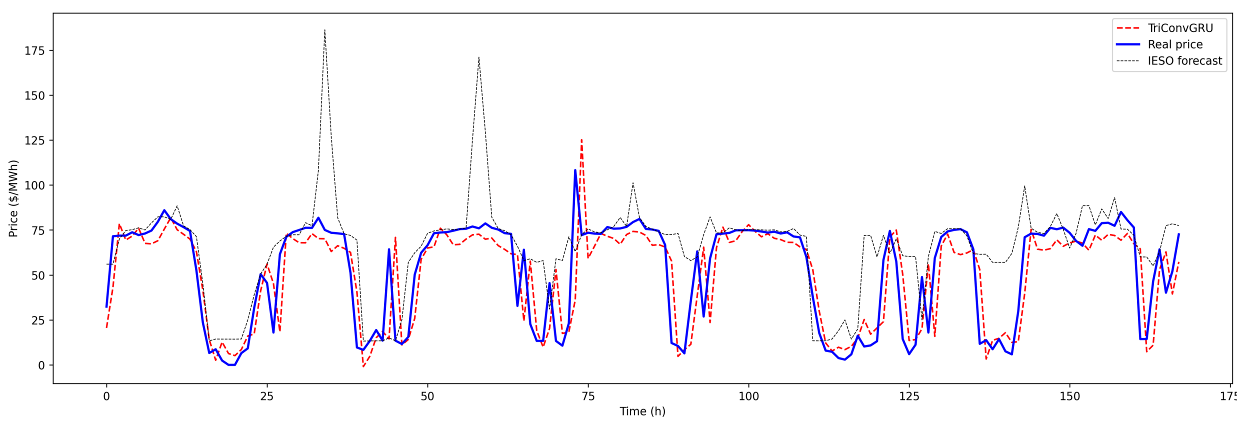
- Compared to the IESO forecast, the univariate proposed model, as the model with the best result, reduced the RMSE error dramatically by 64.11% and 51.41% in the first week, 33.37% and 34.9% in the second week, and 28.42% and 23.08% in the third week



(a) One-step hour forecast for the first week



(b) One-step hour forecast for the second week



(c) One-step hour forecast for the third week

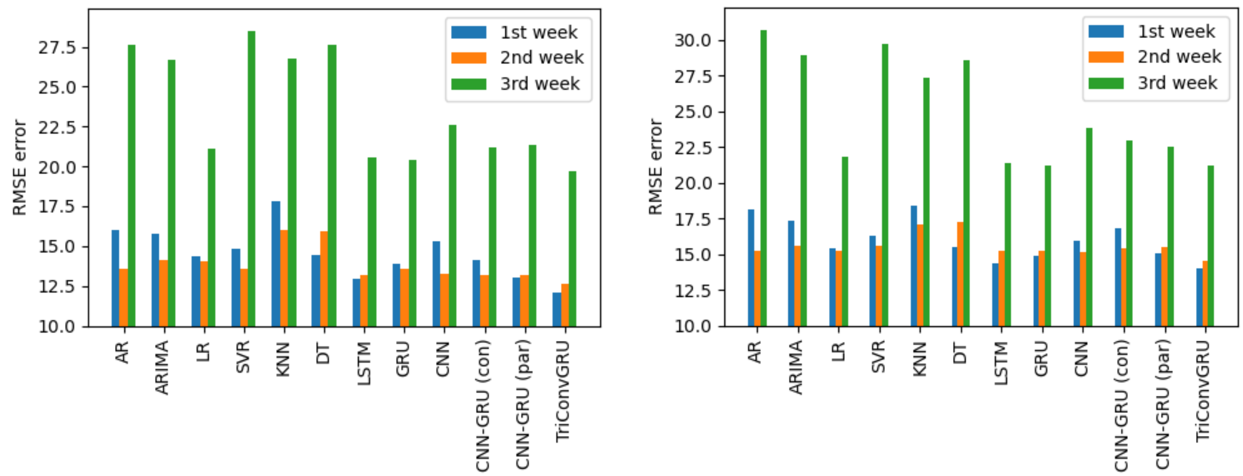
Figure 20: One-step ahead comparison of the TriConvGRU with the IESO forecast and the real price

for two-step and three-step ahead forecasting, respectively. The sensible improvement can be observed in Figures 22a - 22c and Figures 23a - 23c, which show the proposed model captures the ups and downs in sample weeks.

- In two-step and three-step ahead univariate forecasting, better results are obtained by the DL models compared to the statistical and ML models, as depicted in Figure 21. Among DL models, the proposed model represents the best result based on RMSE and MAE metrics. Thus, it is concluded that DL is the best choice for univariate multi-step forecasting. Although all DL models in multi-step forecasting were not outstanding among models in the multivariate setting, some of them, including LSTM and GRU, performed well compared with other models.

- As observed in the results of two-step and three-step ahead forecasting, incorporating variables in most DL and ML models reduced RMSE and MAE errors in the first and second weeks. However, the result for the third week deteriorates while variables are added. For those weeks when the price has fewer fluctuations, the variables help to decrease the error. On the other hand, the week with drastic fluctuations did not reduce the error. It means that the models are more sensitive to fluctuations or noises, and it represents the overfitting problem in these models. However, our proposed model did not show this behavior. We conclude that for multi-step ahead forecasting, adding variables may not help improve the forecasting power. Also, the multi-step ahead result of the proposed model for univariate forecasting is better than the best results of multivariate forecasting, obtained by LSTM, GRU, and LR.

- Similar to one-step ahead, although the proposed complex hybrid model improves the MSE or MAE metrics for two- and three-step ahead forecasting, the simple consecutive and parallel hybrid models did not impact the results of basic DL models, including LSTM and GRU, and thus a network with more parameters is required.



(a) Two-step ahead forecasting

(b) Three-step ahead forecasting

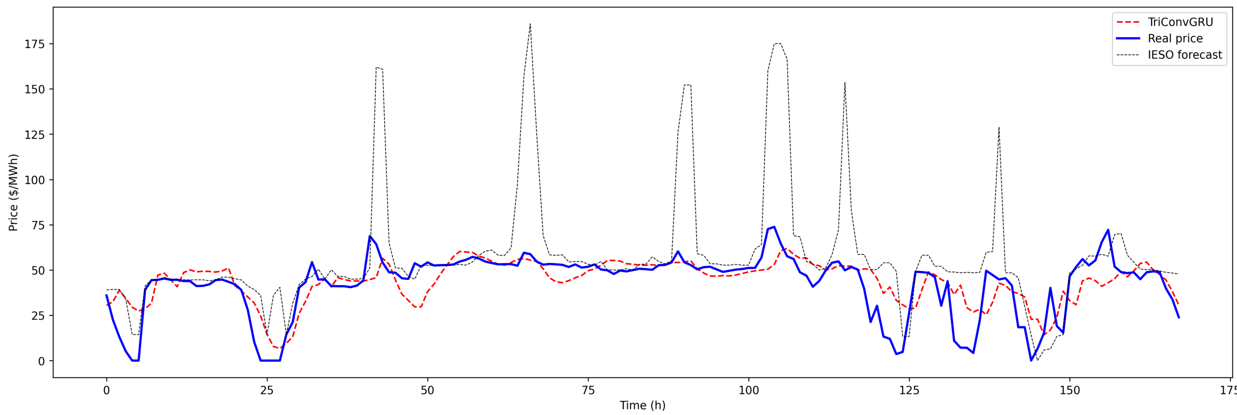
Figure 21: Comparison of RMSE metrics for univariate multi-step forecasting

3.4. Ablation Study

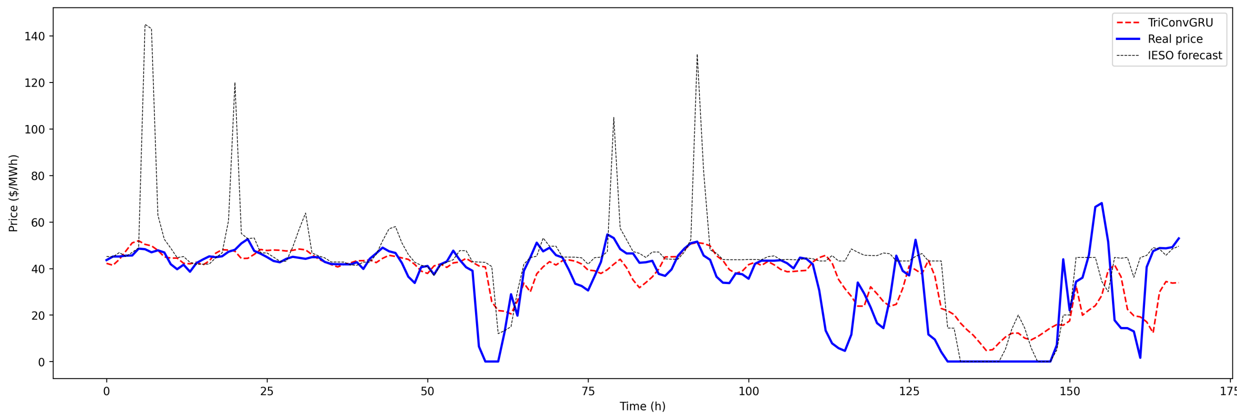
To illustrate the effectiveness of the TriConvGRU-TCG Model, a thorough ablation study was carried out. The study involves removing a component of a DL architecture to assess its effect on the overall network, proposed by Meyes et al. (2019). In this study, one stream or component of the model was removed at a time and the performance of the model was evaluated without that stream. The specific ablated models used in this study are as follows:

- Model A: One-head ConvGRU without down-sampling
- Model B: One-head ConvGRU with a $\frac{1}{4}$ down-sampling
- Model C: One-head ConvGRU with a $\frac{1}{8}$ down-sampling
- Model D: Two-head ConvGRU with no and $\frac{1}{4}$ down-samplings
- Model E: Two-head ConvGRU with no and $\frac{1}{8}$ down-samplings
- Model F: Three-head GRU without convolution layers

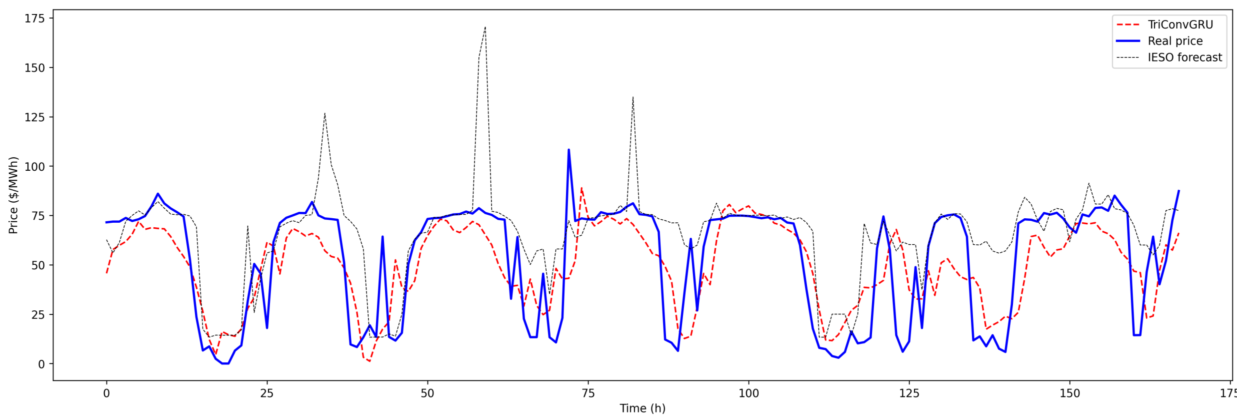
Table 13 provides a summary of the findings of the ablation study. All comparisons were carried out using univariate forecasting methods because of their superiority, as mentioned in previous sections. Models A, B, and C pertain to each head of the proposed model. The comparison of these models illustrates that while the kernel size of average pooling layers increases in downsampling, the result worsens because the downsampled input data with larger kernel size learn longer-term trends or the lower frequency of price patterns.



(a) Two-step hour forecasting for the first week

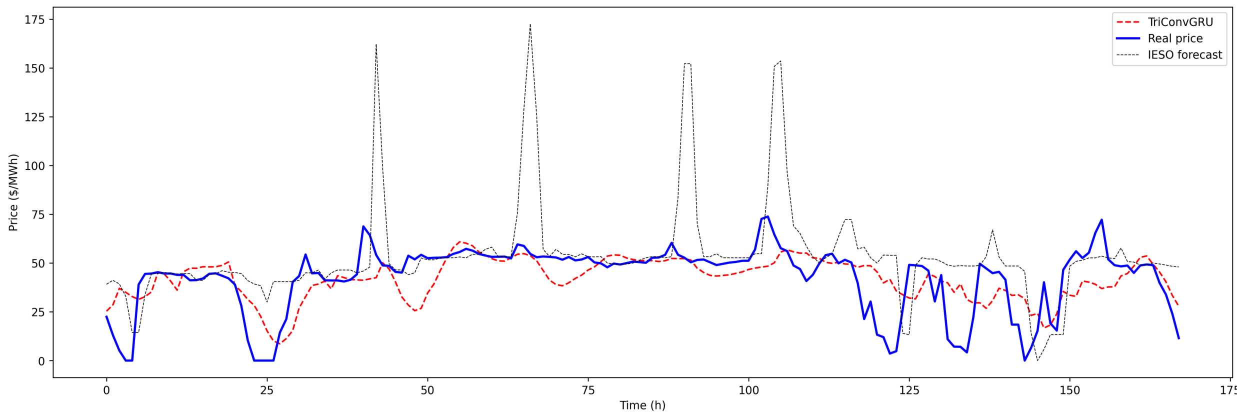


(b) Two-step hour forecasting for the second week

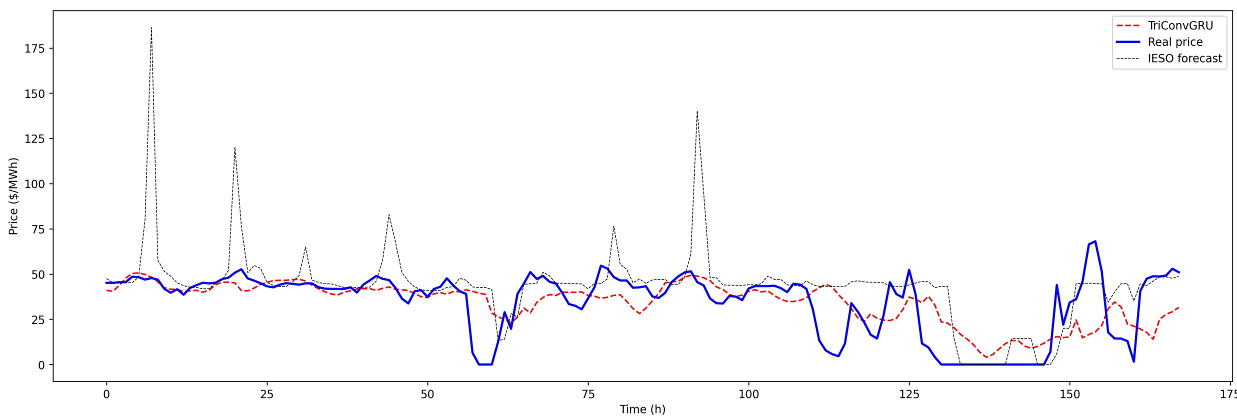


(c) Two-step hour forecasting for the third week

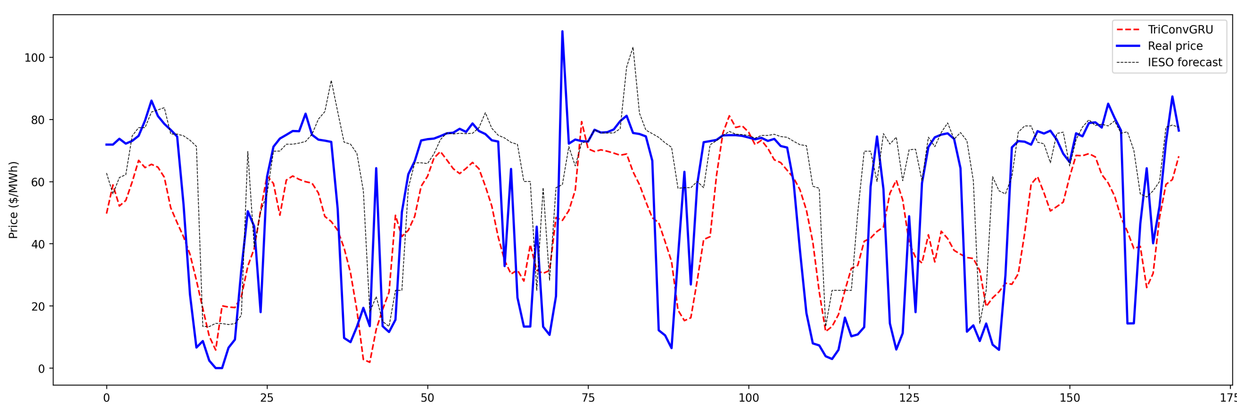
Figure 22: Two-step ahead comparison of the TriConvGRU with the IESO forecast and the real price



(a) Three-step hour forecasting for the first week



(b) Three-step hour forecasting for the second week



(c) Three-step hour forecasting for the third week

Figure 23: Three-step ahead comparison of the TriConvGRU with the IESO forecast and the real price

Furthermore, by comparing the proposed model with models D and E, removing one of the three heads of the TriConvGRU-TCG architecture and considering just two heads worsen the model performance. it is notable, the largest deterioration of performance in T+1 and T+2 forecasts occur when a $\frac{1}{4}$ down-sampling head is removed because a $\frac{1}{4}$ down-sampling head concentrates on shorter trends of input price data. Also, by comparing the results of hybrid consecutive and parallel CNN-GRU models with models D and E, we found a noticeable reduction in the errors. In addition, models D and E in most cases outperform the simple GRU and LSTM whereas hybrid consecutive and parallel CNN-GRU models can not outperform them before. Thus, this result confirms the inclusion of pooling layers in a hybrid model architecture could be effective to reduce errors and learn price patterns because the low frequency of price trends can be captured and incorporated into forecasts, leading to being less sensitive to noises and more concentrated on price trends. As seen in our proposed model, the multiple frequencies of price patterns are learned together, resulting in a superior output. Also, based on model F, removing the CNN layers from the architecture of TriConvGRU-TCG increases the errors in all forecasted results, demonstrating the importance of the CNN layer in extracting the most valuable time features.

3.5. Model Confidence Set Test

In this section, we assess the forecasting accuracy of models statistically, including our proposed model (TriConvGRU) and baseline models, in the univariate setting. The result of the MCS test is reported in Table 14. Notably, we assume that α equals 5% and the loss function used in the test is the **MAE**. We reject the null hypothesis if the p-value of the model comparisons is less than α ($p < \alpha$) at the given significance level. Subsequently, we remove weak models from the model set using the elimination rule. This process is repeated until the null hypothesis cannot be rejected, leading to the creation of a superior set of models.

With respect to Table 14, during the first week, only our proposed model was retained in the superior set, while other models were eliminated in the MCS procedure. In the second

Table 13: Experimental results of the ablation study

Model		27 Jan 2022 - 2 Feb 2022		28 Feb 2022 - 7 Mar 2022		12 May 2022 - 19 May 2022	
		RMSE	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
Model A	T+1	9.750	6.825	8.842	5.771	17.442	12.599
	T+2	13.575	9.942	13.443	9.228	20.795	16.395
	T+3	15.378	11.545	15.156	10.593	21.947	18.079
	Average	12.901	9.437	12.480	8.531	20.061	15.691
Model B	T+1	11.940	8.934	11.211	7.605	18.285	11.920
	T+2	14.314	10.902	13.901	9.867	20.771	16.188
	T+3	16.414	12.464	15.489	10.965	21.741	**17.460**
	Average	14.223	10.767	13.534	9.479	20.266	15.189
Model C	T+1	15.760	11.485	14.209	10.349	24.824	20.776
	T+2	16.261	11.938	15.238	11.097	24.979	20.899
	T+3	16.324	11.944	16.041	11.404	24.147	20.516
	Average	16.115	11.789	15.163	10.950	24.650	20.730
Model D	T+1	8.236	5.527	8.583	5.267	17.931	**11.688**
	T+2	12.171	8.675	13.300	9.258	20.747	16.164
	T+3	14.167	**10.273**	15.477	10.671	21.615	17.530
	Average	11.525	8.158	12.453	8.399	20.098	15.127
Model E	T+1	8.547	5.696	8.747	5.357	18.952	13.361
	T+2	12.285	9.029	13.285	9.345	21.673	16.969
	T+3	**13.989**	10.351	14.649	**10.058**	22.559	18.492
	Average	11.607	8.359	12.227	8.253	21.061	16.274
Model F	T+1	8.768	5.811	8.772	5.696	17.002	11.962
	T+2	12.501	9.297	13.154	9.099	20.343	15.993
	T+3	14.439	11.293	15.339	10.811	21.765	18.002
	Average	11.903	8.800	12.422	8.535	19.703	15.319
Proposed	T+1	**8.177**	**5.393**	**8.106**	**5.070**	17.676	11.879
	T+2	**12.091**	**8.581**	**12.663**	**8.721**	**19.658**	**15.559**
	T+3	14.033	10.333	**14.534**	10.093	**21.153**	17.778
	Average	**11.434**	**8.102**	**11.768**	**7.961**	**19.496**	**15.072**

* **Note: $T+1$, $T+2$, and $T+3$ represent one-step, two-step, three-step ahead forecasting, respectively. Green and Bold numbers indicate the minimum error.**

Table 14: Result of the MCS test for all forecasting models

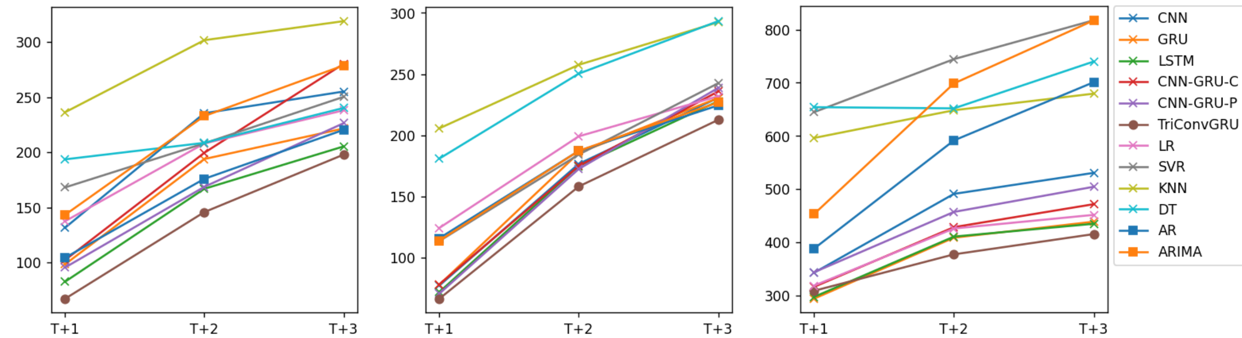
Model	**Week 1**		**Week 2**		**Week 3**	
	p_value	Rank	p_value	Rank	p_value	Rank
AR	x	x	x	x	x	x
ARIMA	x	x	x	x	x	x
LR	x	x	x	x	x	x
SVR	x	x	x	x	x	x
DT	x	x	x	x	x	x
KNN	x	x	x	x	x	x
LSTM	x	x	1.0000	3	**1.0000**	**1**
GRU	x	x	0.0510	6	1.0000	3
CNN	x	x	0.8166	4	x	x
CNN_GRU_C	x	x	0.1342	5	0.1208	5
CNN_GRU_P	x	x	1.0000	2	0.4306	4
TriConvGRU	**1.000**	**1**	**1.0000**	**1**	1.0000	2
Number of eliminated models	11		6		7	

* **Note: The symbol x shows that a model was eliminated via the MCS procedure, and bold numbers indicate that a model achieved the first rank in the superior model set.**

week, all DL models remained in the model set, and our proposed model was ranked first in the set. Similarly, in the third test set, all DL models except for the CNN model remained in the superior model set, and our proposed model was ranked second in the set. The results from the second and third weeks indicate that DL models outperform ML and statistical models.

3.6. Stability Test of the Proposed Model

In electricity markets, prices are prone to significant fluctuations because of market factors. Thus, models that exhibit a high level of stability are considered capable of providing accurate forecasting (Sun and Li, 2020). Hence, the variance of forecasting errors is utilized, Eq. 18, to measure the stability of our proposed model (TriConvGRU), as well as baseline models. Because of the superiority of the average error for the univariate setting, stability test results for univariate models are depicted in Figure 24. Clearly, the TriConvGRU-TCG model yields the lowest variance of forecasting errors compared to baseline models and demonstrates superior robustness in comparison with the other models. Additionally, the variance value tends to increase since the forecasting horizon increases in all models.



(a) For the first week (b) For the second week (c) For the third week

Figure 24: Stability test results

4. Conclusion

In our study, a novel DL model, called TriConvGRU, based on GRU and CNN models, has been designed and assessed empirically using Ontario electricity market prices. Also, a comparative study among statistical, ML, and DL models was conducted to evaluate forecasting methods in Ontario Market. The proposed model is composed of three consecutive hybrid CNN-GRU models, combined in parallel with different input data generated by average pooling layers at the beginning of each stream, except for the first stream. We implemented models in two different settings, including univariate and multivariate with various external variables. Previous prices, electricity load, generation, import and export, and weather data are considered exogenous variables in our study to determine whether independent variables improve forecasts or not.

Based on the performance metrics results, the proposed model in the univariate setting shows promising results. In terms of multivariate and univariate settings, we conclude that incorporating variables into models does not improve forecasting results of DL and ML models because the inclusion of low-correlated features increases the complexity of models and thus overfitting problems happened. Another finding of the research relates to the superiority of DL models compared with statistical and ML models in two different settings. However, similar to the result concluded by Lehna et al. (2022), employing the simple hybrid DL models, both parallel and consecutive, did not enhance the forecasting efficacy, and it is concluded that the hybrid network with more parameters is required to improve the results of simple DL models, e.g., LSTM and GRU. As the final finding, our proposed model outperforms IESO forecasts in both univariate and multivariate settings, and it represents the least error compared with baseline models because the model can learn multiple frequencies of price patterns concurrently by including pooling layers.

In future research, it may be beneficial to integrate the use of filter decomposition techniques, e.g., variational mode decomposition (VMD) (Dragomiretskiy and Zosso, 2013) and empirical mode decomposition (EMD) (Flandrin et al., 2004), into a DL model for the EPF problem. This could potentially improve the efficiency of the forecasting model. One approach to implementing this hybridization method could be based on the work of Zhang et al. (2022). They propose a hybrid LSTM-Attention model with the VMD filter for forecasting coal prices. State-of-the-art forecasting models relying on attention mechanisms could also provide gains for the time-series forecasting problems (Du et al., 2020). For instance, transformer-based models can be considered either on their own (L'Heureux et al., 2022) or hybridized with CNN layers (Shen and Wang, 2022) to further improve forecasting power.

Table A.15: Parameter spaces tuned with a grid search for mentioned models

Algorithm	Parameter	Parameter Space
ARIMA	p	1, 2, 3
	d	0, 1
	q	0, 1, 2, 3
SVR	C	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1
	epsilon	0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1
	kernel	'linear', 'rbf'
DT	max_depth	5, 10, 20, 30, 50, 100
KNN	n_neighbors	5, 10, 15, 20
	weights	'uniform', 'distance'
	leaf_size	30, 60, 90, 120
	p	1, 2
Deep Learning models	hidden_layer	16, 32, 64, 128
	Kernel_size	5, 7, 9
	LR	0.001, 0.0001
	Optimizer	'Adam', 'SGD'

Appendix A. Hyperparameter Spaces

References

J. Abellán, C. J. Mantas, and J. G. Castellano. A random forest approach using imprecise probabilities. *Knowledge-Based Systems*, 134:72–84, 2017.

A. F. Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

H. Akoglu. User's guide to correlation coefficients. *Turkish journal of emergency medicine*, 18(3):91–93, 2018.

C. Albanese, H. Lo, and S. Tompaidis. A numerical algorithm for pricing electricity derivatives for jump-diffusion processes based on continuous time lattices. *European Journal of Operational Research*, 222(2):361–368, 2012.

M. Alhussein, K. Aurangzeb, and S. I. Haider. Hybrid cnn-lstm model for short-term individual household load forecasting. *Ieee Access*, 8:180544–180557, 2020.

N. Amjady and M. Hemmati. Energy price forecasting-problems and proposals for such predictions. *IEEE Power and Energy Magazine*, 4(2):20–29, 2006.

P. Atănăsoae, R. D. Pentiuc, and E. Hopulele. Considerations Regarding the Negative Prices on the Electricity Market. In *Multidisciplinary Digital Publishing Institute Proceedings*, volume 63, page 26, 2020.

R. Baldick, R. Grant, and E. Kahn. Theory and application of linear supply function equilibrium in electricity markets. *Journal of regulatory economics*, 25(2):143–167, 2004.

A. Baldominos, Y. Saez, and P. Isasi. Evolutionary design of convolutional neural networks for human activity recognition in sensor-rich environments. *Sensors*, 18(4):1288, 2018.

C. Batlle and J. Barquín. A strategic production costing model for electricity market price analysis. *IEEE Transactions on Power Systems*, 20(1):67–74, 2005.

M. Benini, M. Marracci, P. Pelacchi, and A. Venturini. Day-ahead market price volatility analysis in deregulated electricity markets. 3:1354–1359 vol.3, 2002.

S. Borenstein, J. Bushnell, and C. R. Knittel. Market power in electricity markets: Beyond concentration measures. *The Energy Journal*, 20(4), 1999.

F. Canova. Vector autoregressive models: specification, estimation, inference, and forecasting. *Handbook of applied econometrics volume 1: Macroeconomics*, pages 53–110, 1999.

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.

W. H. Chung, Y. H. Gu, and S. J. Yoo. District heater load forecasting based on machine learning and parallel cnn-lstm attention. *Energy*, 246:123350, 2022.

A. J. Conejo, M. A. Plazas, R. Espinola, and A. B. Molina. Day-ahead electricity price forecasting using the wavelet transform and arima models. *IEEE transactions on power systems*, 20(2):1035–1042, 2005.

J. Contreras, R. Espinola, F. J. Nogales, and A. J. Conejo. Arima models to predict next-day electricity prices. *IEEE transactions on power systems*, 18(3):1014–1020, 2003.

T. Cover. Estimation by the nearest neighbor rule. *IEEE Transactions on Information Theory*, 14(1):50–55, 1968.

A. S. Dagoumas, N. E. Koltsaklis, and I. P. Panapakidis. An integrated model for risk management in electricity trade. *Energy*, 124:350–363, 2017.

E. De Alba. Constrained forecasting in autoregressive time series models: A bayesian analysis. *International Journal of Forecasting*, 9(1):95–108, 1993.

Z. Deng, C. Liu, and Z. Zhu. Inter-hours rolling scheduling of behind-the-meter storage operating systems using electricity price forecasting based on deep convolutional neural network. *International Journal of Electrical Power & Energy Systems*, 125:106499, 2021.

F. Divina, M. Garcia Torres, F. A. Goméz Vela, and J. L. Vazquez Noguera. A comparative study of time series forecasting methods for short term electric energy consumption prediction in smart buildings. *Energies*, 12(10):1934, 2019.

K. Dragomiretskiy and D. Zosso. Variational mode decomposition. *IEEE transactions on signal processing*, 62(3):531–544, 2013.

S. Du, T. Li, Y. Yang, and S.-J. Horng. Multivariate time series forecasting via attention-based encoder–decoder framework. *Neurocomputing*, 388:269–279, 2020.

A. Ehsan and Q. Yang. State-of-the-art techniques for modelling of uncertainties in active distribution network planning: A review. *Applied energy*, 239:1509–1523, 2019.

N. Elsayed, A. S. Maida, and M. Bayoumi. Deep gated recurrent and convolutional network hybrid model for univariate time series classification. *arXiv preprint arXiv:1812.07683*, 2018.

A. Eydeland and K. Wolyniec. *Energy and power risk management: New developments in modeling, pricing, and hedging*, volume 97. John Wiley & Sons, 2002.

B. Farsi, M. Amayri, N. Bouguila, and U. Eicker. On short-term load forecasting using machine learning techniques and a novel parallel deep lstm-cnn approach. *IEEE Access*, 9:31191–31212, 2021.

P. Flandrin, G. Rilling, and P. Goncalves. Empirical mode decomposition as a filter bank. *IEEE signal processing letters*, 11(2):112–114, 2004.

A. Fragkioudaki, A. Marinakis, and R. Cherkaoui. Forecasting price spikes in european day-ahead electricity markets using decision trees. In *2015 12th International Conference on the European Energy Market (EEM)*, pages 1–5. IEEE, 2015.

M. W. Gardner and S. Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15): 2627–2636, 1998.

E. Guerci, M. A. Rastegar, and S. Cincotti. Agent-based modeling and simulation of competitive wholesale electricity markets. In *Handbook of power systems II*, pages 241–286. Springer, 2010.

X. Guo, Q. Zhao, D. Zheng, Y. Ning, and Y. Gao. A short-term load forecasting model of multi-scale cnn-lstm hybrid neural network considering the real-time electricity price. *Energy Reports*, 6:1046–1053, 2020.

M. Gürtler and T. Paulsen. Forecasting performance of time series models on electricity spot markets: a quasi-meta-analysis. *International journal of energy sector management*, 2017.

N. Haldrup, F. S. Nielsen, and M. Ø. Nielsen. A vector autoregressive model for electricity prices subject to long memory and regime switching. *Energy Economics*, 32(5):1044–1058, 2010.

P. R. Hansen, A. Lunde, and J. M. Nason. The model confidence set. *Econometrica*, 79(2): 453–497, 2011.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

M. A. Hossain, R. K. Chakrabortty, S. Elsawah, E. M. Gray, and M. J. Ryan. Predicting wind power generation using hybrid deep learning with optimization. *IEEE Transactions on Applied Superconductivity*, 31(8):1–5, 2021a.

M. A. Hossain, R. K. Chakrabortty, S. Elsawah, and M. J. Ryan. Very short-term forecasting of wind power generation using hybrid deep learning model. *Journal of Cleaner Production*, 296:126564, 2021b.

IESO. Introduction to ontario's physical markets. https://www.ieso.ca/-/media/Files/IESO/Document-Library/training/WB-Intro-Ontario-Physical-Markets.ashx, 2014.

W. A. R. Intan Azmira, A. Ahmad, I. Z. Abidin, K. S. Yap, M. N. M. Nasir, and W. R. Upkli. Electricity price prediction with support vector machine and bacterial foraging optimization algorithm for day-ahead model. In *2020 IEEE Student Conference on Research and Development (SCOReD)*, pages 159–164, 2020.

F. Ioannidis, K. Kosmidou, C. Savva, and P. Theodossiou. Electricity pricing using a periodic garch model with conditional skewness and kurtosis components. *Energy Economics*, 95: 105110, 2021.

J. Janczura and R. Weron. An empirical comparison of alternate regime-switching models for electricity spot prices. *Energy economics*, 32(5):1059–1073, 2010.

J. G. Jetcheva, M. Majidpour, and W.-P. Chen. Neural network model ensembles for building-level electricity load forecasts. *Energy and Buildings*, 84:214–223, 2014.

T. A. Johnsen. Demand, generation and price in the norwegian market for electric power. *Energy Economics*, 23(3):227–251, 2001.

N. Khan, I. U. Haq, S. U. Khan, S. Rho, M. Y. Lee, and S. W. Baik. Db-net: A novel dilated cnn based multi-step forecasting model for power consumption in integrated local energy systems. *International Journal of Electrical Power & Energy Systems*, 133:107023, 2021.

J. Lago, F. De Ridder, and B. De Schutter. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. *Applied Energy*, 221:386–405, 2018.

Q. V. Le, N. Jaitly, and G. E. Hinton. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*, 2015.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

M. Lehna, F. Scheller, and H. Herwartz. Forecasting day-ahead electricity prices: A comparison of time series and neural network models taking external regressors into account. *Energy Economics*, 106:105742, 2022.

C. Li, Y. Tao, W. Ao, S. Yang, and Y. Bai. Improving forecasting accuracy of daily enterprise electricity consumption using a random forest based on ensemble empirical mode decomposition. *Energy*, 165:1220–1227, 2018.

C. Li, G. Li, K. Wang, and B. Han. A multi-energy load forecasting method based on parallel architecture cnn-gru and transfer learning for data deficient integrated energy systems. *Energy*, 259:124967, 2022a.

X. Li, C. Liang, Z. Chen, and M. Umar. Forecasting crude oil volatility with uncertainty indicators: New evidence. *Energy Economics*, 108:105936, 2022b.

H. Liu and J. Shi. Applying arma–garch approaches to forecasting short-term electricity prices. *Energy Economics*, 37:152–166, 2013.

T. Liu, G. Xu, P. Cai, L. Tian, and Q. Huang. Development forecast of renewable energy power generation in china and its influence on the ghg control strategy of the country. *Renewable Energy*, 36:1284–1292, 04 2011.

A. L'Heureux, K. Grolinger, and M. A. Capretz. Transformer-based model for electrical load forecasting. *Energies*, 15(14):4993, 2022.

K. Maciejowska, W. Nitka, and T. Weron. Enhancing load, wind and solar generation for day-ahead forecasting of electricity prices. *Energy Economics*, 99:105273, 2021.

R. B. Marimont and M. B. Shapiro. Nearest neighbour searches and the curse of dimensionality. *IMA Journal of Applied Mathematics*, 24(1):59–70, 1979.

J. Mei, D. He, R. Harley, T. Habetler, and G. Qu. A random forest method for real-time price forecasting in new york electricity market. In *2014 IEEE PES General Meeting—Conference & Exposition*, pages 1–5. IEEE, 2014.

A. Meng, P. Wang, G. Zhai, C. Zeng, S. Chen, X. Yang, and H. Yin. Electricity price forecasting with high penetration of renewable energy using attention-based lstm network trained by crisscross optimization. *Energy*, 254:124212, 2022.

R. Meyes, M. Lu, C. W. de Puiseau, and T. Meisen. Ablation studies in artificial neural networks. *arXiv preprint arXiv:1901.08644*, 2019.

M. M. Mukaka. A guide to appropriate use of correlation coefficient in medical research. *Malawi medical journal*, 24(3):69–71, 2012.

M. Mulder and B. Scholtens. The impact of renewable energy on electricity prices in the netherlands. *Renewable energy*, 57:94–100, 2013.

D. Niu, L. Sun, M. Yu, and K. Wang. Point and interval forecasting of ultra-short-term wind power based on a data-driven method and hybrid deep learning model. *Energy*, page 124384, 2022.

I. P. Panapakidis and A. S. Dagoumas. Day-ahead electricity price forecasting via the application of artificial neural network based models. *Applied Energy*, 172:132–151, 2016.

H. Park, J. W. Mjelde, and D. A. Bessler. Price dynamics among us electricity spot markets. *Energy Economics*, 28(1):81–101, 2006.

R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR, 2013.

M. Paschen. Dynamic analysis of the german day-ahead electricity spot market. *Energy Economics*, 59:118–128, 2016.

L. Peng, S. Liu, R. Liu, and L. Wang. Effective long short-term memory with differential evolution algorithm for electricity price prediction. *Energy*, 162:1301–1314, 2018.

A. Pourdaryaei, H. Mokhlis, H. A. Illias, S. H. A. Kaboli, S. Ahmad, and S. P. Ang. Hybrid ann and artificial cooperative search algorithm to forecast short-term electricity price in de-regulated electricity market. *IEEE Access*, 7:125369–125386, 2019.

J. R. Quinlan. Learning decision tree classifiers. *ACM Computing Surveys (CSUR)*, 28(1): 71–72, 1996.

N. Rafizadeh. Negative electricity prices in north america. 2022. doi: https://energie.hec.ca/wp-content/uploads/2022/05/Rapport-detude_2022-2_Rafizadeh.pdf.

A. Rahman, V. Srikumar, and A. D. Smith. Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. *Applied energy*, 212: 372–385, 2018.

R. Rick and L. Berton. Energy forecasting model based on cnn-lstm-ae for many time series with unequal lengths. *Engineering Applications of Artificial Intelligence*, 113:104998, 2022.

T. Rintamäki, A. S. Siddiqui, and A. Salo. Does renewable energy generation decrease the volatility of electricity prices? an analysis of denmark and germany. *Energy Economics*, 62:270–282, 2017.

C. P. Rodriguez and G. J. Anders. Energy price forecasting in the ontario competitive power system market. *IEEE transactions on power systems*, 19(1):366–374, 2004.

S. Saab, E. Badr, and G. Nasr. Univariate modeling and forecasting of energy consumption: the case of electricity in lebanon. *Energy*, 26(1):1–14, 2001.

A. G. Salman, Y. Heryadi, E. Abdurahman, and W. Suparta. Single layer & multi-layer long short-term memory (lstm) model with intermediate variables for weather forecasting. *Procedia Computer Science*, 135:89–98, 2018.

D. C. Sansom, T. Downs, and T. K. Saha. Evaluation of support vector machine based forecasting tool in electricity price forecasting for australian national electricity market participants. *Journal of Electrical & Electronics Engineering, Australia*, 22(3):227–233, 2003.

M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90: 106181, 2020.

L. Shen and Y. Wang. Tcct: Tightly-coupled convolutional transformer on time series forecasting. *Neurocomputing*, 480:131–145, 2022.

A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

N. Son. Comparison of the deep learning performance for short-term power load forecasting. *Sustainability*, 13(22):12493, 2021.

W. Sun and Z. Li. Hourly pm2. 5 concentration forecasting based on mode decomposition-recombination technique and ensemble learning approach in severe haze episodes of china. *Journal of Cleaner Production*, 263:121442, 2020.

I. Sutskever, J. Martens, and G. E. Hinton. Generating text with recurrent neural networks. In *ICML*, 2011.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

T. Terlouw, T. AlSkaif, C. Bauer, and W. van Sark. Multi-objective optimization of energy arbitrage in community energy storage systems using different battery technologies. *Applied energy*, 239:356–372, 2019.

U. Ugurlu, I. Oksuz, and O. Tas. Electricity price forecasting using recurrent neural networks. *Energies*, 11(5):1255, 2018.

F. vom Scheidt, H. Medinová, N. Ludwig, B. Richter, P. Staudt, and C. Weinhardt. Data analytics in the electricity sector–a quantitative and qualitative literature review. *Energy and AI*, 1:100009, 2020.

Y. Wang, N. Zhang, and X. Chen. A short-term residential load forecasting model based on lstm recurrent neural network considering weather features. *Energies*, 14(10):2737, 2021.

R. Weron. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International journal of forecasting*, 30(4):1030–1081, 2014.

R. Weron and A. Misiorek. Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models. *International journal of forecasting*, 24(4):744–763, 2008.

W. Xia, W. Zhu, B. Liao, M. Chen, L. Cai, and L. Huang. Novel architecture for long short-term memory used in question classification. *Neurocomputing*, 299:20–31, 2018.

M. Xie, C. Sandels, K. Zhu, and L. Nordström. A seasonal arima model with exogenous variables for elspot electricity prices in sweden. In *2013 10th International Conference on the European Energy Market (EEM)*, pages 1–4. IEEE, 2013.

H. Yang and K. R. Schell. Ghtnet: Tri-branch deep learning network for real-time electricity price forecasting. *Energy*, 238:122052, 2022a.

H. Yang and K. R. Schell. Qcae: A quadruple branch cnn autoencoder for real-time electricity price forecasting. *International Journal of Electrical Power & Energy Systems*, 141:108092, 2022b.

Z. Yang, L. Ce, and L. Lian. Electricity price forecasting by a hybrid model, combining wavelet transform, ARMA and kernel-based extreme learning machine methods. *Applied Energy*, 190:291–305, 2017.

H. Zareipour. Short-term electricity market prices: A review of characteristics and forecasting methods. *Handbook of networks in power systems I*, pages 89–121, 2012.

J. Zarnikau, C. K. Woo, S. Zhu, and C. Tsai. Market price behavior of wholesale electricity products: Texas. *Energy Policy*, 125:418–428, 2019.

G. P. Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

K. Zhang, H. Cao, J. Thé, and H. Yu. A hybrid model for multi-step coal price forecasting using decomposition technique and deep learning algorithms. *Applied Energy*, 306:118011, 2022.

S. Zhang, L. Yao, A. Sun, and Y. Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):1–38, 2019.

L. Zhao, C. Su, H. Huang, Z. Han, S. Ding, and X. Li. Intrusion detection based on device-free localization in the era of iot. *Symmetry*, 11(5):630, 2019.

M. Zhou, Z. Yan, Y. Ni, and G. Li. An arima approach to forecasting electricity price with accuracy improvement by predicted errors. In *IEEE Power Engineering Society General Meeting, 2004.*, pages 233–238. IEEE, 2004.

F. Ziel and R. Weron. Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks. *Energy Economics*, 70:396–420, 2018.