Received April 19, 2020, accepted May 7, 2020, date of publication May 14, 2020, date of current version May 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2994444

Locational Marginal Price Forecasting Using Deep Learning Network Optimized by Mapping-Based Genetic Algorithm

YING-YI HONG[1], (Senior Member, IEEE), JONATHAN V. TAYLAR[2], AND ARNEL C. FAJARDO[3], (Member, IEEE)

[1]Department of Electrical Engineering, Chung Yuan Christian University, Taoyuan 32023, Taiwan
[2]Department Computer Engineering, Technological Institute of the Philippines, Quezon City 1109, Philippines
[3]Manuel L. Quezon University, Quezon City 1101, Philippines

Corresponding author: Ying-Yi Hong (yyhong@ee.cycu.edu.tw)

This work was supported in part by the Institute of Nuclear Energy Research, Taiwan, under Grant 108-B-015, and in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 109-3116-F-006-019-CC1 and Grant MOST 108-2221-E-033-023.

ABSTRACT The convolutional neural network (CNN) is commonly used in visual recognitions and classifications. However, CNN can also be applied as a forecaster that can extract features from spatiotemporal data. This paper proposes a 24h ahead electricity price forecasting method, which integrates CNN with an evolutionary algorithm and utilizes spatiotemporal data. The optimal structure of the CNN network for the locational marginal price (LMP) forecasting was obtained using a genetic algorithm (GA). A gene mapping scheme was initially encoded to represent the search space and the process of selection, mutation, and crossover eliminated structures that did not satisfy the validation fitness function and then competitive individuals were generated. The evolution process uses the root mean square error (RMSE) as the validation fitness function, which is optimzed by training the created CNN network. The proposed gene mapping scheme can be used to design an optimal CNN structure once the mapping between gene binary bits and parameters/hyperparameters of CNN is given. Day-ahead LMP and demand datasets from Pennsylvania-New Jersey-Maryland (PJM) power market were used to demonstrate the evolutionary capability of the proposed method and the finding of optimal CNN structures. Each studied dataset was grouped into 4 subsets corresponding to various seasonal characteristics (different types of situations in real life). Experimental results revealed that the proposed GA-CNN always yielded a higher forecasting accuracy and lower error rates than other forecasting methods.

INDEX TERMS Convolutional neural network, deep learning, electricity price forecasting, genetic algorithm, locational marginal price.

I. INTRODUCTION

Electricity pricing has been a crucial indicator of all transactions in the power market since the reformation of the power industry [1], [2]. When prices are high, sellers should produce electricity to the pool market or buyers should use their own generating facilities. Sellers/buyers can change their bidding schemes to maximize their benefit and protect themselves from financial risk. These sellers (producers) and buyers (consumers) depend greatly on electricity price forecasting for planning and managing price risks. Many factors must be considered in forecasting the price of electricity in the power markets. These factors include demand, supply, weather, and other variables that are related to the fuel markets [3], [4]. The volatility of electricity prices and large errors in the use of forecasting techniques in other markets should also be considered [5]. In the PJM power market, the locational marginal price (LMP) is used to charge the price bought and sold energy. LMP involves pricing congestion costs into energy transmission within a regional transmission organization (RTO) and considering bulk power system losses. Authorized by the federal government, PJM is responsible for electricity transmission systems and the operations of the wholesale electricity market in a particular

The associate editor coordinating the review of this manuscript and approving it for publication was Pierluigi Siano.

area. Commonly, LMP aggregates system energy prices, transmission congestion costs, and the cost of marginal losses.

Over the last two decades, recommended price forecasting methods have had varying rates of success; they include statistical methods and artificial intelligence methods. Statistical methods include autoregressive integrated moving averages (ARIMA) [6]–[8] the similar days' method [3], generalized autoregressive conditional heteroskedasticity (GARCH) [8],[9] and other autoregression (AR) methods. These methods are linear predictors, which are only accurate when data vary slowly. The effectiveness of these methods is uncertain because of their limited ability to capture the nonlinear variation of electricity prices and rapid variations in price signals [10]. Artificial intelligence (AI) methods include artificial neural networks (ANN) [11], deep learning networks [12]–[15] evolutionary computing methods, and hybrid models or combinations of at least two methods [14], [16]–[19]. AIs can approximate any multivariate function to an expected degree of accuracy by adjusting weightings during updates and they can be used to extract the complex features of electricity prices. Hence, these AI methods have higher forecasting accuracy than statistical methods. Different evolutionary computing methods such as genetic algorithm (GA) [20]–[22] and particle swarm optimization (PSO) [4], [19], [22], [23], were used in conjunction with other algorithms to forecast electricity prices. These combinations represent excellent means of price forecasting because they combine a linear autocorrelation structure with a nonlinear component.

CNN and GA have already been combined in recent studies [24], [25], but these studies have focused on using CNN for classifying images and used evolutionary algorithms to optimize its parameters or to generate the optimal CNN network. Other combinations, such as a combinatorial neural network trained by a stochastic search method [26], were presented to study the datasets from PJM and Spain. The results were compared with those obtained using other methods and proved to be more extensive.

Many researchers have attempted to determine the optimal solution to a particular problem by translating mathematically using a fitness function of specific parameters. Such mathematical methods have become foundational for the various optimizations that are of interest today. Today, deep learning has been used to solve difficult problems, yielding results that are similar to or superior to those obtained by human experts. However, setting the parameters of a deep learning network can be difficult because their values control the learning process and determine the performance of a deep learning network.

Grid search [27]–[29] and random search [28], [30], [31] are traditional techniques of hyperparameter optimization. Both techniques are directed by performance metrics that are evaluated by cross-validation on the training and validation sets. Bayesian optimization [15],[17], [32]–[34] is used to optimize parameters by creating a probabilistic model of the functional mapping from the values of the parameters to the objective that is evaluated on a validation set.

Classical optimization techniques have difficulty in attaining the global optimum owing to their inability to handle nonlinearities and non-convexity. On the other hand, evolutionary algorithms (EA) are members of the family of population-based algorithms, which were developed to find quasi-optimal solutions in any complex search field. Particle Swarm Optimizations (PSO) [4], [19], [22], [23] and Evolutionary Strategy[24], [35] are developed to solve optimization problems of continuous variables without any constraints. Although variants of PSO (such as Discrete PSO) can deal with binary or discrete variables, it is still difficult to deal with inequality constraints (e.g., maximum and minimum limits of a variable). Consequently, an enhanced GA based on mapping encoding is proposed herein because GA can deal with binary variables and inequality constraints efficiently.

A novel hybrid electricity price forecasting method [20] was applied to historical data from the New England area. A set of relevance vector machines (RVM) was used for individual ahead-of-time price forecasting. Individual predictions are combined to form as a linear regression ensemble, whose coefficients are obtained as the solution to a single optimization problem. The solution was found using a micro-genetic algorithm, which yielded the optimized ensemble which provided the final price forecast. Another study used GA [21] to optimize the parameters of the support vector machine (SVM) model, which was used to forecast prices in large power systems using data from the National Electricity Market (NEM) of Australia. Another study [36] used a novel model-based demand response control method to control the use of residential air conditioners and optimally in response to changing day-ahead electricity prices. GA is used to find the optimal solution to indoor air temperature, which was formulated as a nonlinear programming problem. Simulation results showed a reduction in the electricity costs and peak power demands during demand response hours.

This paper concerns multivariate time series forecasting, specifically using LMP spatiotemporal data series. The aim is to forecast the 24h-ahead LMPs at a target location using other related time-series by designing a CNN architecture that is optimized by an evolutionary algorithm. Initially, a gene mapping scheme is designed to represent the CNN structure and connectivity. The advantage of this scheme is its flexibility in creating network structures of various lengths. The main contributions of this paper are as follows.

- Implementation of LMP and demand time-series as "2D images" applied to CNN
- Development of mapping-encoding chromosomes in GA to reduce the length of a bit string
- Optimization of hyperparameters and structural parameters of the CNN using the proposed mapping-encoding-based GA without trial-and-error

The rest of the paper is organized as follows. Section II reviews CNNs and various optimization techniques.

Section III outlines the proposed price forecasting method. Section IV presents the results of the experiments conducted on two datasets, and Section V concludes.

II. BACKGROUND

A. PJM MARKET

Like other products, electricity is bought, sold and traded in wholesale and retail markets. The wholesale market involves the purchasing and selling of power between generators and resellers. Resellers include electricity utility enterprises, power providers, and electricity vendors. The Federal Energy Regulatory Commission (FERC) controls the operations and trades of the wholesale market in most regions of the United States. The wholesale market begins with the generators, which connect to the grid and generate electricity after they have obtained the necessary permits. The electricity that is produced by generators is purchased by entities that will resell it to satisfy end-user demand. The resale entities purchase electricity through markets or contracts with individual sellers. In many cases, utility companies own generation facilities and sell directly to customers. The price of wholesale electricity is known by a buyer and seller from the wholesale market. PJM power market is one of successful power markets, as shown in Fig. 1, in the world.

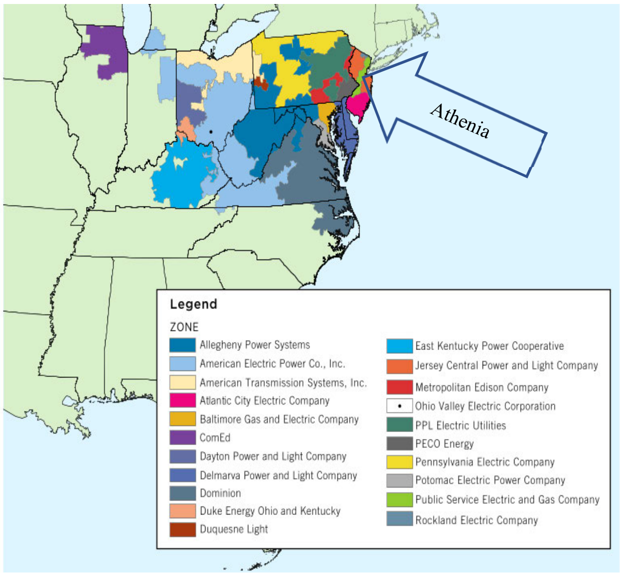

Athenia
Legend
ZONE
Allegheny Power Systems
American Electric Power Co., Inc.
American Transmission Systems, Inc.
Atlantic City Electric Company
Baltimore Gas and Electric Company
ComEd
Dayton Power and Light Company
Delmarva Power and Light Company
Dominion
Duke Energy Ohio and Kentucky
Duquesne Light
East Kentucky Power Cooperative
Jersey Central Power and Light Company
Metropolitan Edison Company
Ohio Valley Electric Corporation
PPL Electric Utilities
PECO Energy
Pennsylvania Electric Company
Potomac Electric Power Company
Public Service Electric and Gas Company
Rockland Electric Company

FIGURE 1. PJM map of 21 zones and target location (Athenia) [37].

As a regional transmission organization (RTO), PJM manages a wholesale electricity market that spans all or part of Delaware, Illinois, Kentucky, Maryland, Michigan, New Jersey, North Carolina, Ohio, Pennsylvania, Tennessee, Virginia, West Virginia and the District of Columbia. Figure 1 shows the map of the 21 zones that comprise the PJM interconnections[37], which are American Transmission Systems, Inc. (ATSI), Atlantic City Electric Company, Baltimore Gas and Electric Company, ComEd (CE), Dayton Power and Light Company (DAY), Duke Energy Ohio and Kentucky (DEOK), Dominion (DOM), American Electric Power (AEP) Co., Inc., Allegheny Power Systems (APS), Delmarva Power and Light Company, Duquesne Light, East Kentucky Power Cooperative (EKPC), PECO Energy, Jersey Central Power and Light Company, Metropolitan Edison Company, Ohio Valley Electric Corporation, PPL Electric Utilities, Pennsylvania Electric Company, Potomac Electric Power Company, Public Service Electric and Gas Company (PSEG), and Rockland Electric Company. Athenia, which is the target location can be located in PSEG.

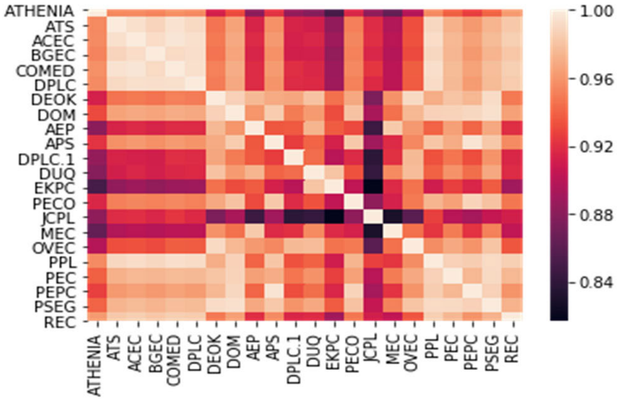


FIGURE 2. Correlation heatmap of Athenia and the 21 zones of the PJM market.

B. DATASET

The proposed method uses two datasets, which were obtained from the PJM website [38]. The dimension of the first dataset is 22 × 2184, corresponding to the 21 zonal prices of the zones that make up the PJM interconnection and the LMP of the target location, "Athenia". The value of 2184 is the number of hours in a season; the total number of hours from December 1, 2017 to November 30, 2018, is 8736. This dataset was used to construct 2-dimensional spatiotemporal forms as inputs to the 2D CNN, creating a multivariate time-series, which yields multiple variables in zones and at a location at a single time; multiple channels per time-series input are thus obtained. Figure 2 displays the correlation heatmap of the target location (Athenia) and the 21 zones of the PJM power market. The LMPs of the 21 zones were considered in Dataset 1 because the target location has strong correlation scores within [0.84, 1) for all PJM zones. In addition to the aforementioned 22 LMPs, Dataset 2 also includes the demands. These demands in only eight zones (AEP, ATSI, APS, CE, DAY, DEOK, DOM, and EKPC) are available on the PJM website.

Feature scaling was applied to both datasets to yield values between zero and one. Let x, x_{max}, x_{min} and x_{new} be the datum, maximum, minimum and scaled datum, respectively. The formula for the feature scaling is shown in (1) [39] using the min-max normalization as follows.

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (1)$$

The PJM power market is very well-developed. Thus, there are neither missing data nor outliers in the studied datasets. In case that missing data or abnormal values occur in the datasets, a general ''data cleansing'' process can be conducted as follows: (i) interpolation, (ii) prior experience (persistence; duplication of prior data), or (iii) usage of the average value from neighboring locations (zones).

C. CONVOLUTIONAL NEURAL NETWORK

Recurrent neural networks (RNNs) [13], particularly those with a long-short term memory unit (LSTM) [40], [41] are the preferred single time-series forecasters. The effectiveness of these networks can be evaluated in terms of the recurrent connections, which permit the network to access all historical time series values. However, CNN can have multiple convolution layers, in which filters are applied by skipping elements in the input, allowing the network to grow exponentially. This process allows the network to access a wide range of historical data, like RNN. The convolutional structure of CNN and its number of trainable weights is small, making it more efficient in training and forecasting than RNN.

CNN is a biologically inspired means of deep learning, which has shown promise in solving classification problems, such as image recognition, segmentation, object detection, and time-series classification and prediction. CNN comprises a sequence of convolutional layers whose outputs are connected to a local region in the input by sliding a filter along with the input and computing the dot product of the input and the filter at each point. The convolutions replace the weighted sums from the neural network. The structure enables the model to identify specific patterns in the input data. Accordingly, CNN discovers filters that define repeating patterns in the series and use them to forecast future values. The layered structure of the CNN works efficiently on noisy series by removing the noise in each succeeding layer and extracting only the meaningful patterns. A feature map is generated when the input is convolved with the filter in each layer. Unlike typical neural networks, the values in the output feature map share similar weights, so all of the nodes in the output detect exactly similar patterns. This process reduces the number of learnable parameters and increases the efficiency of training and learning in every layer.

Although the literature presents many versions of CNN, all use similar algorithms. The building blocks of CNN are the convolutional layer, the pooling layer, and the fully connected layer. Figure 3 presents an example of CNN architecture. Convolutional layers comprise a rectangular grid of neurons so the input layer or the previous layer is a rectangular grid of neurons, in which all of the weights of the rectangular section are the same for all neurons. The convolutional layer generates an image convolution of the preceding layer in which weights denote the convolution filter. The pooling layer receives small rectangular blocks from the convolutional layer and samples them, yielding a single output from the block. Pooling takes the average or maximum operation of the neurons in the block. A fully connected layer (forecasting layer herein) receives all neurons in the preceding layer (which could be a pooling, convolutional or a fully connected layer) and connects them to each of its neurons. Fully connected layers are one-dimensional whereas convolutional layers and pooling layers are generally two-dimensional.

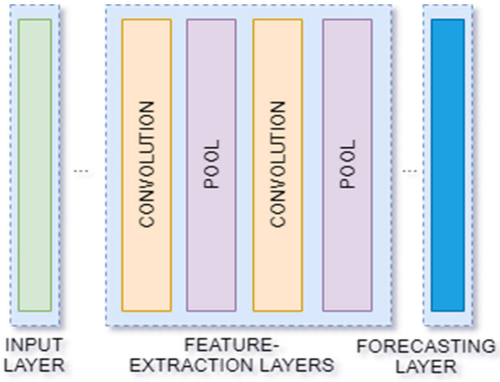


FIGURE 3. Example of CNN architecture.

The convolutional layer performs kernel convolution in which a small matrix of numbers, known as kernel or filter, is passed over the image or the rectangular grid, to convert it based on the values of the filter. This filter outputs new matrices called feature maps. The values of feature maps are calculated using (2) [42] as follows.

$$G[m,n] = (f * h)[m,n] = \sum_{j}\sum_{k} h[j,k]f[m-j,n-k] \tag{2}$$

where f is the input and h is the kernel, and m and n are the indices of the rows and columns of the result matrix.

In this work, these hyperparameters (kernel size, number of kernels, pooling size and dropout) and structural parameters (number of layers) of CNN are encoded to the GA to characterize CNN.

III. PROPOSED METHOD

A. IMPLEMENTATION OF GA

Tuning the parameters and hyper-parameters of CNN is laborious if a trial-and-error approach is used. This was performed in our early experiments and in other existing papers, too. However, this brute-force search is time-consuming and cannot guarantee the optimum. This motivates the authors to apply GA to find the optimal structure of CNN.

GA is an optimization technique for solving complex problems by iteratively considering various probable solutions. In this work, GA is used to find the optimal parameters of CNN, which is used as a standalone forecaster for 24h-ahead spatiotemporal data. This goal is to demonstrate the efficiency with which GA searches the solution space using the following phases; (a) gene mapping - population initialization, (b) genetic operation, and (c) chromosome evaluation.

Beginning with the initialization of the population size and the number of generations, GA applies a series of evolutionary operators until it obtains the best architecture of CNN for LMP forecasts. First, a population is initialized to have a predefined random size using gene encoding or gene mapping, as will be discussed in Sec. III.C. Throughout evolution, the fitness of each individual, which generates a specific CNN architecture is evaluated from the given dataset.

In the subsequent process, the parent individuals are selected according to fitness, and generate new offspring by the application of genetic operators. The selection operator selects from the current population the individuals who survive in the next generation. Therefore, the current population includes the generated offspring population and the parent population. The evolution continues until the predefined maximum number of generations is reached.

B. MAPPING-BASED ENCODING

At the beginning of the GA process, an initial population of chromosomes (binary strings; individuals) is encoded in the solution search space. Gene mapping enables the algorithm to locate a binary string in a genome pool. The gene mapping corresponds to the considered parameters of CNN and assessed in terms of fitness. Initially, a chromosome is set to a fixed length of 22 bits (the genotype), as shown in Table 1; its corresponding values are the phenotypes and are measurable. The hyperparameters/parameters to be optimized are the number of convolutions (bits 1 to 3 from the left), the filter size for layer 1 (bits 4 to 7), the kernel size (bits 8 and 9), the pool size (bits 10 and 11), the filter size for layer 2 (bits 12 to 15), the filter size for layer 3 (bits 16 to 19) and the dropout size (bits 20 to 22).

TABLE 1. Gene mapping in proposed method.

No. of Convolutions		Filter no. 1, 2 and 3		Kernel size		Pooling size		Dropout	
Bit	Value	Bit	Value	Bit	Value	Bit	Value	Bit	Value
000	0	0000	32	00	1	00	0	000	0.0
001	1	0001	64	01	3	01	1	001	0.1
010	2	0010	96	10	4	10	2	010	0.2
011	3	0011	128	11	5	11	3	011	0.3
100	4	0100	160					100	0.4
101	5	0101	192					101	0.5
110	6	0110	224					110	0.6
111	7	0111	256					111	0.7
		1000	288						
		1001	320						
		1010	352						
		1011	384						
		1100	416						
		1101	448						
		1110	480						
		1111	512						

Each combination of binary bits represents a feasible solution to the problem. The number of convolutions was set from zero to seven, but in the manual experiments without GA, the maximum number of convolutions that yielded satisfying results was only three. The experiments revealed that if GA used more than three convolutions, then a resource exhausted error occurred, and the updated parameters cannot converge at the maximum number of iterations. Accordingly, a return function was set whenever the GA selected (zero, four, five, six or seven) convolutions to prevent errors in the program. However, for filter numbers (with an interval range of 32), kernel size, pool size, and dropout size, the function returns the equivalent values in the gene mapping.

C. GENETIC OPERATIONS

The genetic operators that are used in this work are similar to those presented in many other works. They include selection, mutation, and crossover, which are used iteratively until the convergence criteria are satisfied.

The deap.tools module, for which is available online [43], [44] is used to execute the selection operation, which performs tournament selection and returns a list that contains references to selected individuals. This operator allows the best individual to be randomly selected from the individuals that participate in each tournament. The parameters include the number of individuals to be chosen, which is denoted by k and is set to 1 in this study, a list of individuals to select from, the number of individuals that participate in the tournament, and the attribute of the individuals that are used in the selection criterion.

The crossover operation performs a two-point crossover on individuals in the input sequence. The places of the two individuals are switched and their unique lengths are retained. The parameters specify the two participating individuals in the crossover and a tuple for both individuals is returned. The mutation operator executes a function that applies a Gaussian mutation of the mean (mu) and standard deviation ($sigma$) *to* the input individual. The parameters mu =0, $sigma$ =1, and $indpb$ =0.2 (probability rate) are used herein[43], [44]. These parameters include the individual subject for mutation, the mean or sequence of means in the Gaussian mutation, the standard deviation or sequence of standard deviations in the Gaussian mutation, and the independent probability that an individual is mutated. This operation returns a tuple for one individual.

D. OBJECTIVE FUNCTION AND ACCURACY METRICS

The purpose of GA is to find the optimal hyperparameters/parameters of CNN for the LMP forecasting. Accordingly, GA sets a fitness function for the selected individuals in its selection process. The objective function is RMSE [4], [45], [46], which is obtained from the evaluation of the created CNN, where it returns the minimal values of

RMSE in (3).

$$RMSE = \sqrt{\frac{1}{N}\sum\nolimits_{i=1}^{N}(y_i - \hat{y}_i)^2} \quad (3)$$

where $\hat{y}$ is the forecasted value; y is the actual value; and $\bar{y}$ denotes the mean of the forecasted values. The accuracy and error rates are evaluated after the optimal CNN architecture is obtained using the following performance metrics:RMSE, R-squared (coefficient of determination) [2], [47], [48] and mean absolute percentage error (MAPE) [1], [4], [26]; the latter two are defined in (4) and (5). R-squared quantifies the linear correlation between the measured and correlated values and trendline reliability. Perfect correlation yields a value of unity; thus, an R-squared value closer to one indicates more accurate forecasting. The objective function is evaluated for each solution that is obtained by the GA algorithm via CNN training. All such solutions are subsequently ranked as the population evolves through several operations, such as selection, crossover, and mutation, to optimize the fitness function and yield the final optimal solution.

$$MAPE = \frac{1}{N}\sum\nolimits_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \quad (4)$$

$$R^2 = 1 - \frac{\sum(\hat{y}_i - y_i)^2}{\sum(y_i - \bar{y})^2} \quad (5)$$

E. IMPLEMENTATION OF THE PROPOSED METHOD

In this paper, Tensorflow was used to generate the code for CNN and importing the modules from Keras, including Conv2D, AveragePooling2D, Dropout, Activation, Flatten, Dense, and ZeroPadding2D. The DEAP library or Distributed Evolutionary Algorithm for Python was used to create the algorithm and to integrate GA and CNN. The DEAP library facilitates the mutation, selection, and evaluation operations. The algorithmic steps are detailed as follows.

The algorithm starts with initializing values such as the number of generations (n), population size (p), hall of fame (hof), crossover rate (cxpb), and mutation rate (mutpb). The initialized population will be saved into a variable (*pop*). The population will be trained using the dataset and its architecture and will be saved on another variable (*pop_cnn*). Using the validation set, *pop_cnn* will be validated and will select the lowest RMSE. The lowest RMSE will be saved to a variable (*best_cnn*). This process will continue until the maximum iteration is reached. The inner loop of the algorithm comprises the mutation and selection process of the genetic algorithm, where 2 CNN architectures with the lowest RMSE will be selected from *pop_cnn.* The 2 selected architectures will generate an offspring using crossover and saves the offspring to a variable (*pop_offspring*). An architecture will be selected from *pop_cnn* and will be mutated to produce a mutated architecture, which will be saved in *pop_offspring.* The mutation and crossover processes continue and update the population. When the maximum iteration is reached, the best CNN architecture will be selected and validate its accuracy and error rates.

```
Proposed GA- CNN Algorithm
Input:    Number of Generations ($n$)
          Population size ($p$)
          Hall of Fame ($hof$)
          Crossover rate ($cxpb$)
          Mutation rate ($mutpb$)
Output:   24-hour ahead LMP forecasting accuracy
Initialize population ($p$)
Save the initialize population to $pop$
for $i = 0$ to $n$ do
  train $pop$ using the training dataset
  save CNN architecture in $pop\_cnn$
  validate $pop\_cnn$ using the validation dataset
  select lowest $RMSE$ and save it to $best\_cnn$
  for $a = 1$ to x do
    from $pop\_cnn$, select 2 CNN architecture ($Cj$ and
    $Ck$) which has the $lowest\ RMSE$
    Generate offspring for $Cj$ and $Ck$ using the assigned
    $cxpb$ rate
    save offspring to $pop\_offspring$
    select a CNN structure from $pop\_cnn$
    mutate the CNN structure using $mutpb$ rate and
    generate a mutated CNN architecture
    save the CNN architecture to $pop\_offspring$
  end for
  update $pop \ll pop\_cnn + pop\_offspring$
end for
Select best CNN architecture
Validate accuracy using R-Squared and error rates using
RMSE and MAPE
```

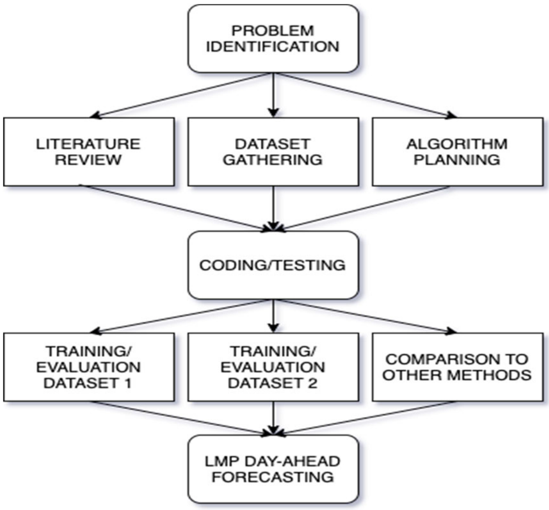

PROBLEM IDENTIFICATION
LITERATURE REVIEW
DATASET GATHERING
ALGORITHM PLANNING
CODING/TESTING
TRAINING/ EVALUATION DATASET 1
TRAINING/ EVALUATION DATASET 2
COMPARISON TO OTHER METHODS
LMP DAY-AHEAD FORECASTING

FIGURE 4. Flowchart of methodology in conducted study.

Figure 4 shows the flowchart of the methodology in the conducted study. The conducted research started with identifying the crucial problem in the power market.

Next, a thorough review of existing papers was performed concerning the problem and possible solutions. After the possible solution was identified, datasets were collected, and an algorithm was developed to avoid the demerits of existing methods and enhance the merits of the proposed method. The next phase was to develop the codes following the presented algorithm. These codes were tested for initial runs of the model. To verify the accuracy of the proposed method, training and evaluation were carried out using 2 datasets. The same datasets were also studied by other methods to compare the performance obtained by all methods. Finally, the proposed method was used to forecast the 24-hour day-ahead LMP for the target location.

The proposed 4 CNNs will be used for forecasting LMPs in different seasons. Because CNN has the capability of learning, the trained CNN using the old data can be retrained using the new data periodically (daily, weekly or monthly) in the corresponding season next year. Updating the parameters and hyperparameters of the CNN still utilizes the RMSE in (3) as an objective function, despite the positive or negative error. Every 24 LMPs and demands in the specified zones and target location will be stored for the subsequent updates of the CNN. The current parameters and hyperparameters of the CNN will serve as initial conditions for the next updates of the CNN.

F. COST-BENEFIT ANALYSIS

There are fixed and variable costs involved in developing the forecasting system in a real working environment. The fixed cost consists of the expenses for hardware: (1) a computer (e.g., an Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz with 256 GB of installed RAM and an NVIDIA GV100GL (Tesla V100 DGXS 32GB) graphics card herein) and (2) the computer network infrastructure for data transmission. The fixed cost also includes the expenses for software: (1) computer code based on the proposed method and (2) the computer network system. The variable cost comprises the cost of maintenance and operation of the above hardware/software as well as personnel salaries. The studied data are free of charge for open access real-time data in the PJM power market. The developed software may be set up using Python with the Keras library and Tensorflow as the backend. TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources. Keras is a high-level neural network interface. Both have no licensing costs.

The forecasted LMP can be served as a crucial reference for buyers/sellers in the power market to develop their bidding strategies. Accordingly, the benefit in the implementation of a forecasting system is the increasing profit due to selling the electricity or decreasing payment owing to purchasing the electricity.

After the above future expected costs and benefits are estimated, they can be converted into a present value amount with a discount rate and the net present value to achieve the cost-benefit study [49], [50].

IV. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed algorithm, a series of experiments on LMP forecasts was performed. Specifically, two datasets are used to demonstrate the effectiveness of the proposed CNN optimized by GA. These two datasets are similar. As described in Sec. II.B, the first dataset consists of 21 zonal prices and target (Athenia) LMP time series. The second dataset comprises the aforementioned 22 price time-series and eight demand time-series.

In all experiments, the following parameters were used; $cxpb = 0.9$ and $mutpb = 0.05$, where $cxpb$ is the crossover rate and $mutpb$ is the mutation rate [43], [44]. The number of binary bits is 22, as indicated in Table 1. The number of populations is set to 110, which is five times the number of binary bits, and the number of generations is 50. For CNN, the parameters in the optimizations provided in Table 1 are as follows; (1) number of convolutions, (2) number of filters for layer 1, (3) kernel size, (4) pool size, (5) number of filters for layer 2, (6) the number of filters for layer 3, and (7) dropout ratio. Table 2 presents the initial parameters of CNN that are used in finding the optimal parameters.

TABLE 2. Initial Setting for Training, Validation and Evaluation.

Parameter/ Hyperparameter	**Values**
Training & validation data	80% (64%Training/16% Validation)
Test Data	20%
Activation function	Rectified Linear Unit (ReLu)
Optimizer	Adam(lr=0.1,beta_1=0.9,beta_2=0.999)
Number of epochs for GA (training)	50
Number of epochs for evaluation	500 with an early stopping mechanism
Batch size, Strides, Dense	4, 1, 50

A. DATASET 1: LMP ONLY

To demonstrate the ability of GA to find the optimal parameters of CNN for forecasting LMPs at Athenia, experiments with the first dataset, covering four seasons, were performed. The first dataset comprises a pure time-series of locational marginal prices on all seasons, including holidays. The target location ''Athenia'' is in the PSEG zone of the PJM interconnection. Using the class deap.tools.Logbook, the evolution process generates a list of statistics during the optimization process; these include the minimum, maximum, mean and standard deviation.

Figure 5 presents the minimum fitness statistics for all four seasons, that are obtained from the optimization process using the first dataset. Notably, the minimum fitness in the winter converged to 0.043 after 20 generations until the 50th generation. However, minimum fitness in spring converged as early as the fourth generation to 0.083 until the final iteration. The minimum fitness in the summer converged by the fifth generation to 0.029 until the final iteration. Finally, the

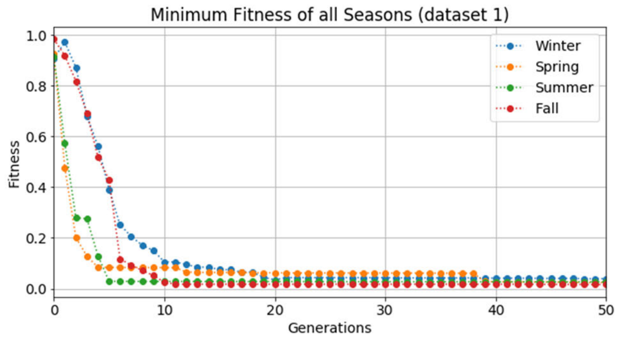


FIGURE 5. Minimum fitness of the proposed model obtained from the first dataset.

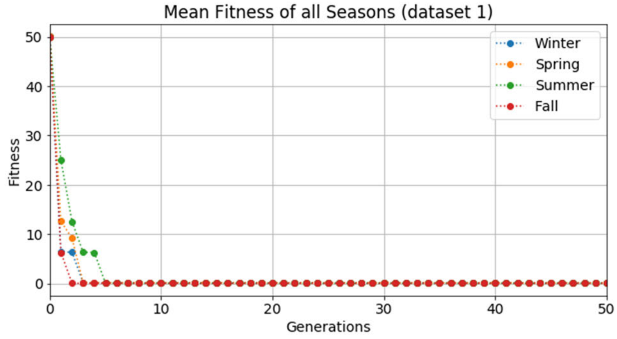


FIGURE 6. The mean fitness of the proposed model obtained from the first dataset.

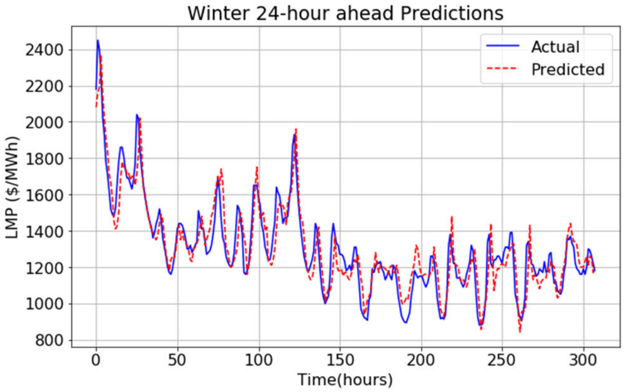


FIGURE 7. 24h-ahead LMP forecasting for winter (dataset 1).

minimum fitness in the fall converged after the tenth generation to 0.017 until the last iteration.

The average or mean fitness values in all the seasons converged in early iterations, as seen in Fig. 6. In winter and spring, the mean fitness converged to 0.1 and 0.16, respectively, after the third generation; in summer, it converged by the fourth generation to 0.057, and in fall, it converged after the 2nd generation to 0.04.

Figures 7-10 plot the LMP forecastings in all seasons using the first dataset; the blue lines represent the actual LMP, and the red dashed lines represent the predicted LMP. The results suggest minimal error rates in all seasons, as the predicted values are very close to the actual values.

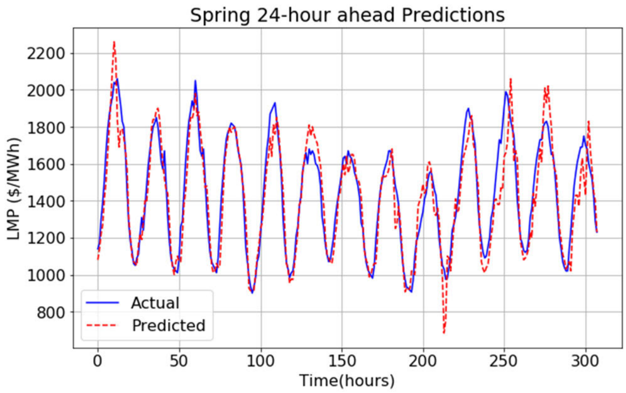


FIGURE 8. 24h-ahead LMP forecasting for spring (dataset 1).

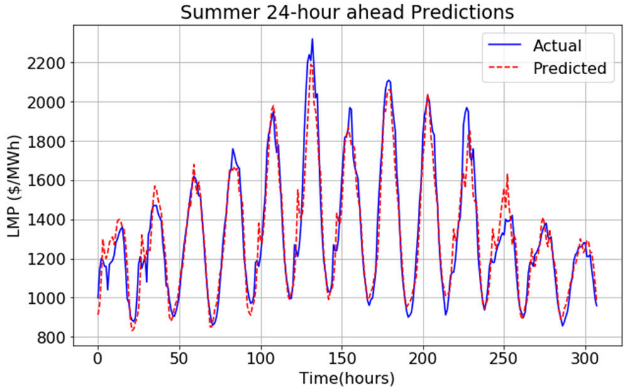


FIGURE 9. 24h-ahead LMP forecasting for summer (dataset 1).

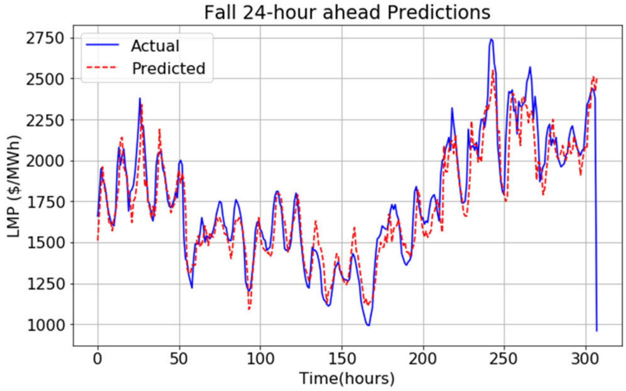


FIGURE 10. 24h-ahead LMP forecasting for fall (dataset 1).

B. DATASET 2: LMP AND DEMAND

The second dataset comprises the 21 zonal price time-series in the first dataset and additional eight historical day-ahead zonal demand values, yielding a 2D spatiotemporal data. Figures 11 and 12 present the minimum and mean fitness statistics of the proposed model for the second dataset.

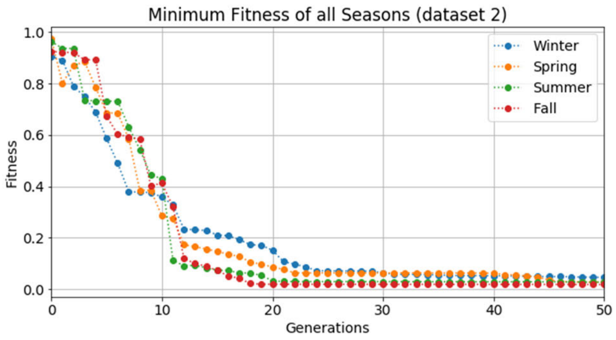


FIGURE 11. Minimum fitness of the proposed model obtained using the second dataset.

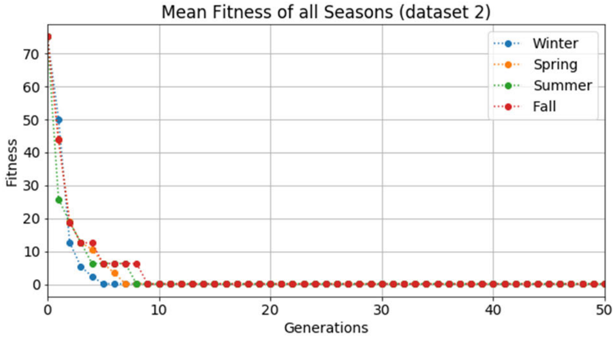


FIGURE 12. The mean fitness of the proposed model obtained using the second dataset.

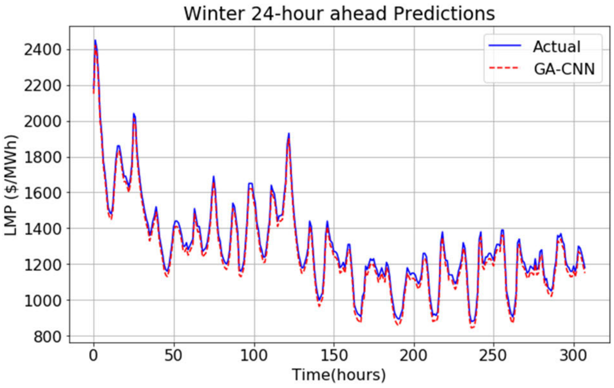


FIGURE 13. 24h-ahead LMP forecasting for winter using the second dataset.

The fitness values converge smoothly in 50 iterations. Figures 13-16 plot predictions for all seasons using the second dataset; the accuracy is much better than that achieved with the first dataset. Figure 17 is the scatter plot indicating the relationship between the actual and predicted values, which is strong and positive.

C. COMPARISON OF RESULTS OBTAINED BY DATASETS 1 AND 2

Tables 3 (a) and 3 (b) present the MAPE, RMSE and R-Squared values obtained by the proposed GA-based CNN for Datasets 1 and 2, respectively. Dataset 2 leads to much lower MAPE and RMSE, and higher R-Squared values than Dataset 1. Restated, the predictions based on Dataset 2 are much better than those based on Dataset 1.

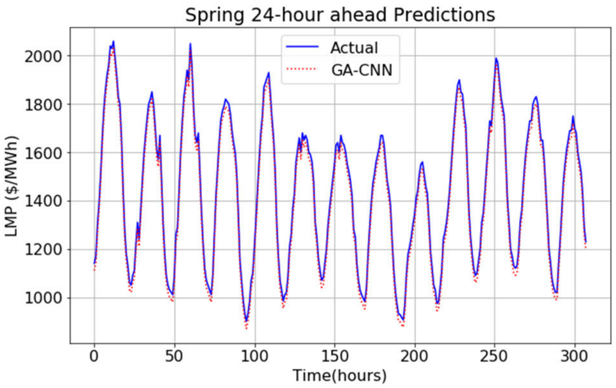


FIGURE 14. 24h-ahead LMP forecasting for spring using the second dataset.

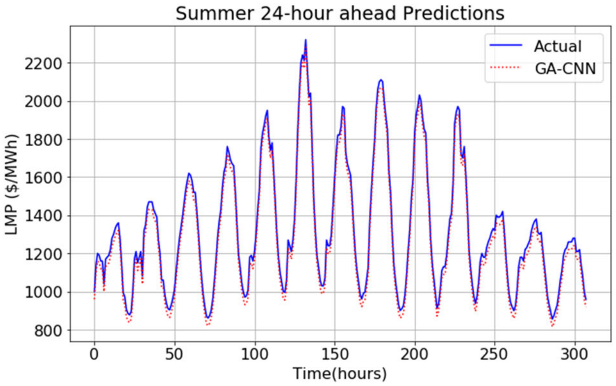


FIGURE 15. 24h-ahead LMP forecasting for summer using the second dataset.

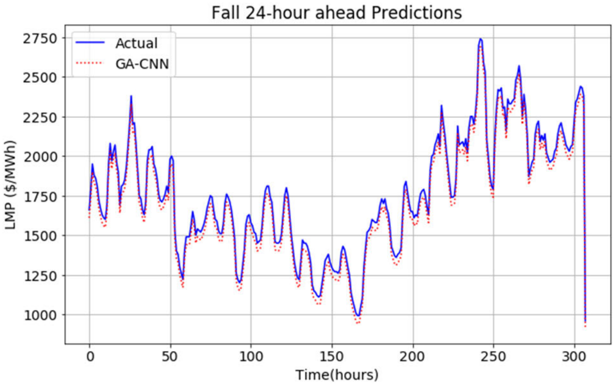


FIGURE 16. 24h-ahead LMP forecasting for fall using the second dataset.

The optimal CNN architectures that were obtained using datasets 1 and 2 vary with the seasons, which are presented in Table 4 (a) and Table 4 (b), respectively. Figures 18 and 19 present the effect of RMSE and R-Squared

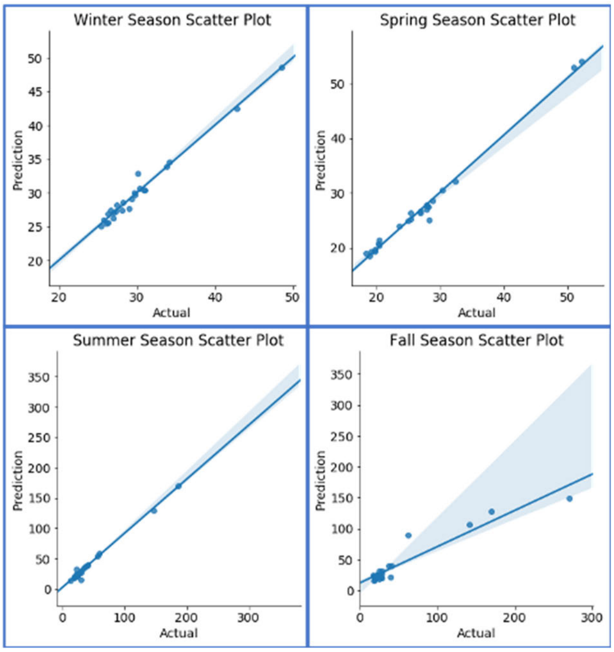


FIGURE 17. Strong positive relationship between actual and predicted values in all seasons using the second dataset.

TABLE 3. (a) Accuracy and Error Rates using Proposed Method (Dataset 1). (b) Accuracy and Error Rates using Proposed Method (Dataset 2).

Seasons	MAPE	RMSE	R-Squared
Winter	4.220	0.0243	0.877
Spring	4.539	0.0069	0.894
Summer	4.666	0.0079	0.897
Fall	4.971	0.0288	0.887

(a)

Seasons	MAPE	RMSE	R-Squared
Winter	3.817	0.0236	0.901
Spring	3.633	0.0086	0.904
Summer	3.584	0.0115	0.908
Fall	3.975	0.0069	0.892

(b)

values of the optimal CNN architecture obtained by dataset 2 on the prediction time horizons, respectively. The distribution of error shows that the RMSE values increase while the R-squared values decrease throughout the 24 hours. Therefore, the one hour-ahead prediction is more accurate than the 24 hour-ahead predictions.

Figure 20 presents an example of an optimal CNN network generated from the proposed method for the winter season, which is described in Table 4 (b). The figure presents the output shape for each layer of the CNN that has three convolutional layers and three pooling layers.

D. COMPARISON OF PROPOSED METHOD WITH OTHER METHODS

The proposed method was compared to other forecasting methods in recent literature. The same datasets were used to test their accuracy and percentage errors in terms of R-squared, RMSE, and MAPE. The compared forecasting methods were long short-term memory (LSTM) [46], support vector machine (SVM) [51], [52], k-nearest neighbor (KNN)[53], Bayesian ridge regression (BR) [54], decision tree method (DT) [55], multilayer perceptron (MLP) [56], [57] and ARIMA [6], [7]. To ensure fair comparisons, the datasets were prepared similar to those used in the proposed method, in which they were scaled within

TABLE 4. (a) Optimal CNN Architectures Created for Dataset 1. (b) Optimal CNN Architectures Created for Dataset 2.

Seasons	Number of Convolutional layers	Filter size for 1, 2 and 3	Kernel size	Pool size	Dropout
Winter	3	128, 256, 512	3 × 3	3 × 3	0.1
Spring	3	512, 512, 352	3 × 3	3 × 3	0.2
Summer	3	512, 64, 480	3 × 3	2 × 2	0.2
Fall	3	256, 512, 480	3 × 3	2 × 2	0.2

(a)

Seasons	Number of Convolutional layers	Filter size for 1, 2 and 3	Kernel size	Pool size	Dropout
Winter	3	384, 32, 320	3 × 3	1 × 3	0.2
Spring	3	512, 512, 160	3 × 3	3 × 3	0.2
Summer	3	512, 256, 192	3 × 3	1 × 3	0.2
Fall	3	512, 128, 128	3 × 3	1 × 3	0.2

(b)

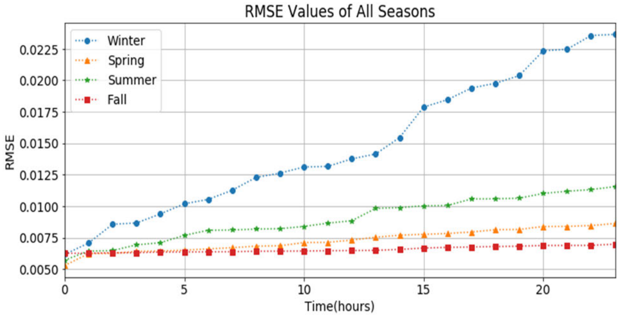


FIGURE 18. Effect of RMSE values of the optimal CNN architectures on prediction time horizons (dataset 2).

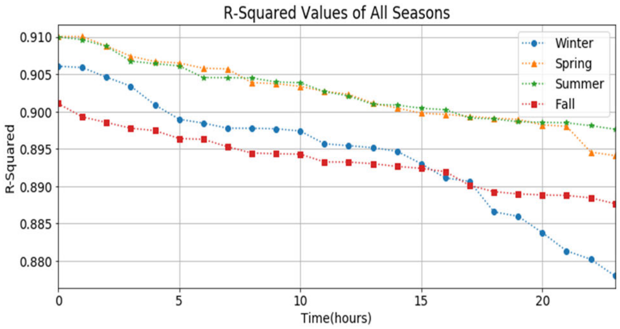


FIGURE 19. Effect of R-Squared values of the optimal CNN architectures on prediction time horizons (dataset 2).

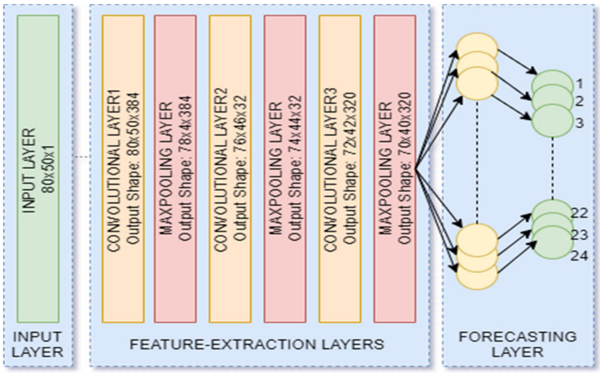

INPUT LAYER 80x50x1
CONVOLUTIONAL LAYER1 Output Shape: 80x50x384
MAXPOOLING LAYER Output Shape: 78x4x384
CONVOLUTIONAL LAYER2 Output Shape: 76x46x32
MAXPOOLING LAYER Output Shape: 74x44x32
CONVOLUTIONAL LAYER3 Output Shape: 72x42x320
MAXPOOLING LAYER Output Shape: 70x40x320
1
2
3
22
23
24
INPUT LAYER
FEATURE-EXTRACTION LAYERS
FORECASTING LAYER

FIGURE 20. Sample CNN architecture optimized by GA.

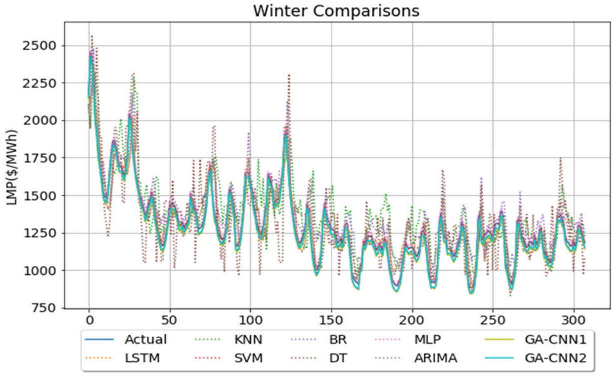


FIGURE 21. Comparisons of all methods using winter data.

0 and 1 using the MinMaxScaler in Sklearn kit. Since the datasets are multivariate time series, they all needed to be arranged and re-shaped as 3D as the input for the different methods. The parameter settings as described in each literature were followed to ensure the best configurations.

Specifically, the parameters of the Adam-optimized LSTM network have a visible layer with one input, a hidden layer with 4 LSTM neurons and an output layer that makes a single value prediction [46]. To make it applicable to our dataset, the output layer has to be designed to have 24 outputs. Different optimizers such as SGD and RMSProp were tested in our experiments but Adam was found to have the best results. The network was trained for 100 epochs and a batch size of 1. The result yielded low MAPE and RMSE values and high R-squared values; however, the proposed method is still better.

For the SVM method, the "linear" kernel function was applied to determine the best model. The input of the SVM is 3D shape. C = 1 and $\gamma = 0.0315$, where C is the penalty parameter and the gamma value influences the support vectors to obtain better scores [51], [52]. The values of C and γ can be easily updated using the Python's Scikit-Learn. The results of SVM are poorer than LSTM.

For the KNN method, only the values of "k" (the number of neighbors) need to be varied. The value of k chosen for testing is from 1 to 25 [53]. The grid search was used to examine the lowest MAPE in determining the best model. The

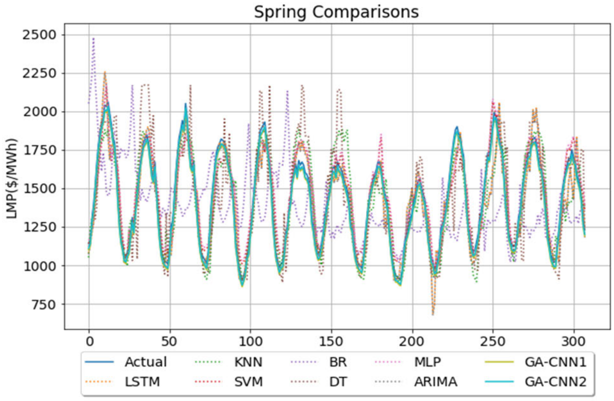


FIGURE 22. Comparisons of all methods using spring data.

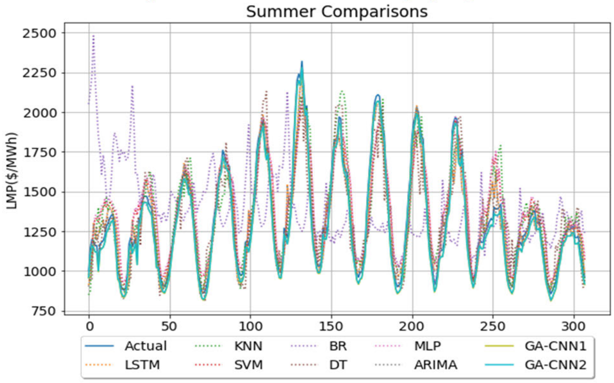


FIGURE 23. Comparisons of all methods using summer data.

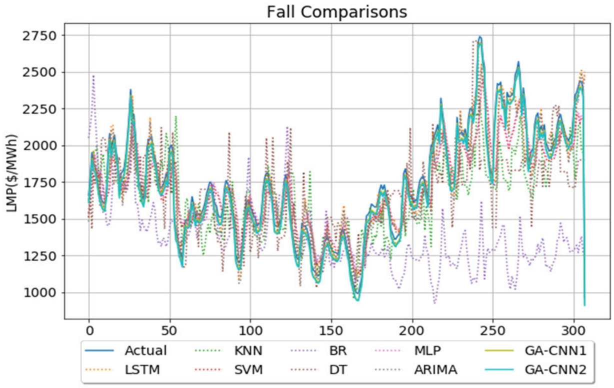


FIGURE 24. Comparisons of all methods using fall data.

results reveal that the KNN method is worse than the SVM method.

For the BR method, the input needs to be 3D shape and the output should be a 24-h LMP forecast. The Scikit-Learn linear model Bayesian Ridge was used to study the dataset. To obtain the best result, Bayesian Optimization was implemented to tune the hyperparameters of the model and optimize the loss function by exploring the underlying distribution [54]. However, only the winter data obtained good results, while the other seasons obtained 100% MAPE values and negative R-squared values.

TABLE 5. (a) Forecasting Comparisons for Dataset 1 (Winter). (b) Forecasting Comparisons for Dataset 1 (Spring). (c) Forecasting Comparisons for Dataset 1 (Summer). (d) Forecasting Comparisons for Dataset 1 (Fall).

Method	MAPE	RMSE	R-Squared
LSTM	7.5113	100.5078	0.8349
SVM	12.6675	156.9419	0.6179
KNN	11.7549	171.1313	0.5427
BR	9.8267	130.4575	0.7208
DTree	9.3104	166.4670	0.6707
MLP	15.5244	160.7134	0.5917
ARIMA	10.0743	135.9632	0.8287
CNN	17.5092	13.2030	0.8296
GA-CNN	**4.2204**	**0.0243**	**0.8773**

(a)

Method	MAPE	RMSE	R-Squared
LSTM	6.9570	108.8507	0.8429
SVM	8.1262	144.9465	0.7507
KNN	9.2113	140.6167	0.7572
BR	100.0000	1.9969	-4.9865
DTree	9.0449	161.9775	0.6945
MLP	16.2757	161.0475	0.5739
ARIMA	15.8422	162.6423	0.7986
CNN	15.1049	10.3673	0.8573
GA-CNN	**4.5393**	**0.0069**	**0.8947**

(b)

Method	MAPE	RMSE	R-Squared
LSTM	6.7307	97.1587	0.8894
SVM	9.0145	118.2749	0.8574
KNN	9.0095	136.8610	0.8297
BR	100.0000	1.8567	-2.9657
DTree	9.1323	145.9471	0.7809
MLP	15.5533	159.6482	0.5708
ARIMA	12.3154	145.2177	0.8286
CNN	12.0017	6.7772	0.8
GA-CNN	**4.6664**	**0.0079**	**0.8972**

(c)

Method	MAPE	RMSE	R-Squared
LSTM	7.2975	156.3834	0.8174
SVM	9.9536	215.2868	0.6567
KNN	12.4402	260.6668	0.5079
BR	100.0000	2.0067	-6.9833
DTree	13.1196	281.7084	0.4075
MLP	16.8825	158.7798	0.5895
ARIMA	14.7609	157.6423	0.8146
CNN	12.1555	7.7835	0.8658
GA-CNN	**4.9715**	**0.0288**	**0.8876**

(d)

TABLE 6. (a) Forecasting Comparisons for Dataset 2 (Winter). (b) Forecasting Comparisons for Dataset 2 (Spring). (c) Forecasting Comparisons for Dataset 2 (Summer). (d) Forecasting Comparisons for Dataset 2 (Fall).

Method	MAPE	RMSE	R-Squared
LSTM	5.8140	97.9071	0.8525
SVM	11.1355	155.4099	0.6284
KNN	10.2729	169.6493	0.5572
BR	8.3477	128.9785	0.7441
DTree	8.1127	165.2693	0.6868
MLP	13.5492	158.7382	0.6048
ARIMA	8.1863	134.0752	0.8404
CNN	15.6012	11.2950	0.8384
GA-CNN	**3.8177**	**0.0236**	**0.9017**

(a)

Method	MAPE	RMSE	R-Squared
LSTM	5.6024	107.4961	0.8675
SVM	6.5942	143.4145	0.7642
KNN	7.7293	139.1347	0.7780
BR	100.0000	0.5179	-4.9246
DTree	7.8472	160.7798	0.7036
MLP	14.3005	159.0723	0.5805
ARIMA	13.9542	160.7543	0.8023
CNN	13.1969	8.4593	0.8625
GA-CNN	**3.6330**	**0.0086**	**0.9049**

(b)

Method	MAPE	RMSE	R-Squared
LSTM	5.3761	95.8041	0.9085
SVM	7.4825	116.7429	0.8763
KNN	7.5275	135.3790	0.8337
BR	100.0000	0.3777	-2.8342
DTree	7.9346	144.7494	0.8099
MLP	13.5781	157.6730	0.6079
ARIMA	10.4274	143.3297	0.8376
CNN	10.0937	4.8692	0.8873
GA-CNN	**3.5843**	**0.0116**	**0.9082**

(c)

Method	MAPE	RMSE	R-Squared
LSTM	5.9429	155.0288	0.8255
SVM	8.4216	213.7548	0.6682
KNN	10.9582	259.1848	0.5121
BR	100.0000	0.5277	-6.3933
DTree	11.9219	280.5107	0.4285
MLP	14.9073	156.8046	0.5985
ARIMA	12.8729	155.7543	0.8223
CNN	10.2475	5.8755	0.8726
GA-CNN	**3.9755**	**0.0070**	**0.8924**

(d)

For the DT method, the model was trained with maximum tree depth of the base learners which was set to 10 [55]. The learning rate, minimum loss reduction, L1 regularization parameter, L2 regularization parameter, and the number of boosted trees were set to 0.09, 0.1, 1e-07, 7 and 800, respectively. To obtain the best parameters, the random grid search from Scikit-Learn was used. However, the results were worse compared to the KNN method.

For the MLP method, the number of hidden layers was set to 2. The parameters in the literature [56] were followed, where the number of epochs = 500, learning rate = 0.001, number of neurons = 10, batch size = 100, and the activation function used is ReLu. The experimental results of MLP are worse than LSTM, SVM and KNN.

Finally, the proposed method was compared to ARIMA method, where the model was set to (5, 1, 0), which sets the lag value = 5 for autoregression, difference order = 1 and moving average model = 0. The experiments in the paper were done using Matlab and R applying it on univariate timeseries. The LSTM results are better than the results obtained from ARIMA. Figures 21-24 compare various methods, including the proposed GA-CNN method for both datasets 1 and 2. The actual LMP values are plotted against the values predicted using the different methods.

Tables 5 (a) to 5 (d) present the percentage of errors and coefficients of determination (R^2) obtained by different methods for Dataset 1, while Tables 4 (a) to 4 (d) are for Dataset 2.

According to the results shown in Tables 5 and 6, the proposed method outperformed the other methods, with the lowest RMSE and MAPE values and the highest R-squared values in all seasons. Besides, the results using Dataset 2 are better than those using Dataset 1.

V. CONCLUSION

This work proposes an efficient method for 24h ahead LMP forecasts using CNN that is optimized using a novel mapping-based GA. The contributions and findings of this paper can be summarized as follows.

- The LMP and demand time-series are preprocessed as 2D (spatiotemporal) data used as inputs to the CNN, thereby, allowing the CNN to successfully capture the spatial and temporal dependencies of the datasets. Thus, additional complex digital signal processing techniques can be avoided but the 2D CNN still can be used.
- The traditional method for tuning hyperparameters and structural parameters of a deep learning network is conducted through brute-force trial and error. The proposed method can find the optimal hyperparameters and structural parameters of CNN by novel mapping-encoding chromosomes in GA to reduce the length of bit-strings without trial and error.
- The proposed method was extensively tested on two datasets from the real-world electricity market of PJM in the United States. From the simulation results, it can be found that hourly demands, in addition to historical LMP data, are also crucial factors as inputs to the 2D CNN to improve the accuracy of 24h ahead LMP forecasting.
- The studied data were grouped into 4 subsets corresponding to various seasonal characteristics (different types of situations in real life). Experiments revealed that the proposed method outperforms the other forecasting methods. These results validate the proposed method.

For future studies and improvement of the proposed method, additional information such as temperature and other factors that affect electricity price forecasting will be considered further to test and validate the method.

At present, the PJM market only provides hourly demands in eight zones (21 zones in total). It can be found Dataset 2 consisting of these eight hourly demands can improve the accuracy of LMP forecasting. Thus, the administrators in the power market shall provide more information for buyers and sellers (participants) to develop their tools (such as LMP forecasters or bidding strategies).

REFERENCES

[1] J. Urpelainen and J. Yang, "Global patterns of power sector reform, 1982–2013," *Energy Strategy Rev.*, vol. 23, pp. 152–162, Jan. 2019.

[2] M. Shahidehpour, H. Yamin, and Z. Li, *Market Operations in Electric Power System*. New York, NY, USA: Wiley, 2003.

[3] P. Mandal, T. Senjyu, N. Urasaki, T. Funabashi, and A. K. Srivastava, "A novel approach to forecast electricity price for PJM using neural network and similar days method," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 2058–2065, Nov. 2007.

[4] 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," *Appl. Energy*, vol. 190, pp. 291–305, Mar. 2017.

[5] L. Wu and M. Shahidehpour, "A hybrid model for day-ahead price forecasting," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1519–1530, Aug. 2010.

[6] E. M. de Oliveira and F. L. Cyrino Oliveira, "Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods," *Energy*, vol. 144, pp. 776–788, Feb. 2018.

[7] D. Bissing, M. T. Klein, R. A. Chinnathambi, D. F. Selvaraj, and P. Ranganathan, "A hybrid regression model for day-ahead energy price forecasting," *IEEE Access*, vol. 7, pp. 36833–36842, 2019.

[8] I. P. Panapakidis, "Energy markets," *Appl. Energy*, vol. 1, no. 3, pp. 1–6, 2018.

[9] C. Bikcora, L. Verheijen, and S. Weiland, "Density forecasting of daily electricity demand with ARMA-GARCH, CAViaR, and CARE econometric models," *Sustain. Energy, Grids Netw.*, vol. 13, pp. 148–156, Mar. 2018.

[10] N. Amjady and M. Hemmati, "Energy price forecasting–problems and proposals for such predictions," *IEEE Power Energy Mag.*, vol. 4, no. 2, pp. 20–29, Mar. 2006.

[11] L. Wang, Z. Zhang, and J. Chen, "Short-term electricity price forecasting with stacked denoising autoencoders," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 2673–2681, Jul. 2017.

[12] B. K. Bose, "Artificial intelligence techniques in smart grid and renewable energy systems—Some example applications," *Proc. IEEE*, vol. 105, no. 11, pp. 2262–2273, Apr. 2017.

[13] U. Ugurlu, I. Oksuz, and O. Tas, "Electricity price forecasting using recurrent neural networks," *Energies*, vol. 11, no. 5, p. 1255, 2018.

[14] J. Lago, F. De Ridder, and B. De Schutter, "Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms," *Appl. Energy*, vol. 221, pp. 386–405, Jul. 2018.

[15] D. Zhang, X. Han, and C. Deng, "Review on the research and practice of deep learning and reinforcement learning in smart grids," *CSEE J. Power Energy Syst.*, vol. 4, no. 3, pp. 362–370, Sep. 2018.

[16] P. Mandal, J. Meng, Srivastava, and R. Martinez, "A novel hybrid approach using wavelet, firefly algorithm, and fuzzy ARTMAP for day-ahead electricity price forecasting," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1041–1051, May 2013.

[17] F. He, J. Zhou, Z.-K. Feng, G. Liu, and Y. Yang, "A hybrid short-term load forecasting model based on variational mode decomposition and long short-term memory networks considering relevant factors with Bayesian optimization algorithm," *Appl. Energy*, vol. 237, pp. 103–116, Mar. 2019.

[18] Y.-Y. Hong and C. L. P. P. Rioflorido, "A hybrid deep learning-based neural network for 24-h ahead wind power forecasting," *Appl. Energy*, vol. 250, pp. 530–539, Sep. 2019.

[19] A. Ghasemi, H. Shayeghi, M. Moradzadeh, and M. Nooshyar, "A novel hybrid algorithm for electricity price and load forecasting in smart grids with demand-side management," *Appl. Energy*, vol. 177, pp. 40–59, Sep. 2016.

[20] M. Alamaniotis, D. Bargiotas, N. G. Bourbakis, and L. H. Tsoukalas, "Genetic optimal regression of relevance vector machines for electricity pricing signal forecasting in smart grids," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 2997–3005, Nov. 2015.

[21] L. M. Saini, S. K. Aggarwal, and A. Kumar, "Parameter optimisation using genetic algorithm for support vector machine-based price-forecasting model in national electricity market," *IET Gener., Transmiss. Distrib.*, vol. 4, no. 1, pp. 36–49, 2010.

[22] J. Zhang, Z. Tan, and S. Yang, "Day-ahead electricity price forecasting by a new hybrid method," *Comput. Ind. Eng.*, vol. 63, no. 3, pp. 695–701, Nov. 2012.

[23] B. Wang, Y. Sun, B. Xue, and M. Zhang, "Evolving deep neural networks by multi-objective particle swarm optimization for image classification," in *Proc. Genetic Evol. Comput. Conf.*, Jul. 2019, pp. 490–498.

[24] B. Wang, Y. Sun, B. Xue, and M. Zhang, "A hybrid differential evolution approach to designing deep convolutional neural networks for image classification," *Lect. Notes Comput. Sci.*, vol. 11320, pp. 237–250, Dec. 2018.

[25] L. Xie and A. Yuille, "Genetic CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 1388–1397.

[26] O. Abedinia, N. Amjady, M. Shafie-khah, and J. P. S. Cataláo, "Electricity price forecast using combinatorial neural network trained by a new stochastic search method," *Energy Convers. Manage.*, vol. 105, pp. 642–654, Nov. 2015.

[27] M. Claesen and B. De Moor, "Hyperparameter search in machine learning," in *Proc. Metaheuristics Int. Conf.*, 2015, pp. 10–14.

[28] P. C. Bhat, H. B. Prosper, S. Sekmen, and C. Stewart, "Optimizing event selection with the random grid search," *Comput. Phys. Commun.*, vol. 228, pp. 245–257, Jul. 2018.

[29] M. Zahid, F. Ahmed, N. Javaid, R. Abbasi, H. Zainab Kazmi, A. Javaid, M. Bilal, M. Akbar, and M. Ilahi, "Electricity price and load forecasting using enhanced convolutional neural network and enhanced support vector regression in smart grids," *Electronics*, vol. 8, no. 2, p. 122, 2019.

[30] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.

[31] Z. Wu, X. Zhao, Y. Ma, and X. Zhao, "A hybrid model based on modified multi-objective cuckoo search algorithm for short-term load forecasting," *Appl. Energy*, vol. 237, pp. 896–909, Mar. 2019.

[32] H. Cheng, X. Ding, W. Zhou, and R. Ding, "A hybrid electricity price forecasting model with Bayesian optimization for german energy exchange," *Int. J. Electr. Power Energy Syst.*, vol. 110, pp. 653–666, Sep. 2019.

[33] J. Lago, F. De Ridder, P. Vrancx, and B. De Schutter, "Forecasting day-ahead electricity prices in europe: The importance of considering market integration," *Appl. Energy*, vol. 211, pp. 890–903, Feb. 2018.

[34] H. Cui and J. Bai, "A new hyperparameters optimization method for convolutional neural networks," *Pattern Recognit. Lett.*, vol. 125, pp. 828–834, Jul. 2019.

[35] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving deep convolutional neural networks for image classification," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 394–407, Apr. 2020.

[36] M. Hu and F. Xiao, "Price-responsive model-based optimal demand response control of inverter air conditioners using genetic algorithm," *Appl. Energy*, vol. 219, pp. 151–164, Jun. 2018.

[37] PJM. (2019). *PJM Maps*. Accessed: Sep. 15, 2019. [Online]. Available: https://www.pjm.com/library/maps.aspx

[38] PJM. *PJM Market*. Accessed: Sep. 5, 2019. [Online]. Available: https://dataminer2.pjm.com/list

[39] V. Sharma and D. Srinivasan, "A hybrid intelligent model based on recurrent neural networks and excitable dynamics for price prediction in deregulated electricity market," *Eng. Appl. Artif. Intell.*, vol. 26, nos. 5–6, pp. 1562–1574, May 2013.

[40] P.-H. Kuo and C.-J. Huang, "An electricity price forecasting model by hybrid structured deep neural networks," *Sustainability*, vol. 10, no. 4, p. 1280, 2018.

[41] K. Wang, X. Qi, and H. Liu, "Photovoltaic power forecasting based LSTM-convolutional network," *Energy*, vol. 189, Dec. 2019, Art. no. 116225.

[42] W. Damelin and S. B., Miller, Jr., *The Mathematics of Signal Processing*, Vol. 48. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[43] F.-A. Fortin, F.-M. De Rainville, M.-A. G. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," *J. Mach. Lang. Res.*, vol. 13, pp. 2171–2175, Jul. 2012.

[44] DEAP. *Distributed Evolutionary Algorithms in Python*. Accessed: Dec. 1, 2019. [Online]. Available: https://deap.readthedocs.io/en/master/

[45] 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*, vol. 7, pp. 125369–125386, 2019.

[46] Z. Chang, Y. Zhang, and W. Chen, "Effective adam-optimized LSTM neural network for electricity price forecasting," in *Proc. IEEE 9th Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Nov. 2018, pp. 245–248.

[47] R. S. Witte and J. S. Witte, *Statistics*. 5th ed. Fort Worth, TX, USA: Harcourt Brace College, 1997.

[48] T. Hong, J. Wilson, and J. Xie, "Long term probabilistic load forecasting and normalization with hourly information," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 456–462, Jan. 2014.

[49] E. M. Carlini and C. Gadaleta, "The 2.0 cost benefit analysis and its application to the National Development Plan 2017," in *Proc. AEIT Int. Annu. Conf.*, Cagliari, Italy, Sep. 2017, pp. 20–22.

[50] V. Padmini, S. Omran, K. Chatterjee, and S. A. Khaparde, "Cost benefit analysis of smart grid: A case study from india," in *Proc. North Amer. Power Symp. (NAPS)*, Morgantown, WV, USA, Sep. 2017, pp. 17–19.

[51] A. Shiri, M. Afshar, A. Rahimi-Kian, and B. Maham, "Electricity price forecasting using support vector machines by considering oil and natural gas price impacts," in *Proc. IEEE Int. Conf. Smart Energy Grid Eng. (SEGE)*, Oshawa, ON, Canada, Aug. 2015, pp. 2–6.

[52] R. A. Swief, Y. G. Hegazy, T. S. Abdel-Salam, and M. A. Bader, "Support vector machines (SVM) based short term electricity load-price forecasting," in *Proc. IEEE Bucharest PowerTech*, Bucharest, Romania, Jun. 2009, pp. 1–5.

[53] A. Zagar, K. Grolinger, M. Capretz, and L. Seewald, "Energy cost forecasting for event venues," in *Proc. IEEE Electr. Power Energy Conf. (EPEC)*, London, U.K., Oct. 2015, pp. 220–226.

[54] K. Pan, W. Shi, X. Wang, and J. Li, "A short-term marginal price forecasting model based on ensemble learning," in *Proc. Int. Conf. Prog. Informat. Comput. (PIC)*, Nanjing, China, Dec. 2017, pp. 81–85.

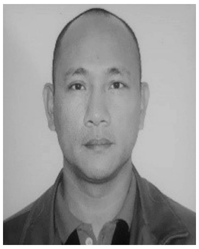
[55] R. K. Agrawal, F. Muchahary, and M. M. Tripathi, "Ensemble of relevance vector machines and boosted trees for electricity price forecasting," *Appl. Energy*, vol. 250, pp. 540–548, Sep. 2019.

[56] R. A. Chinnathambi, S. J. Plathottam, T. Hossen, A. S. Nair, and P. Ranganathan, "Deep neural networks (DNN) for day-ahead electricity price markets," in *Proc. IEEE Electr. Power Energy Conf. (EPEC)*, Toronto, ON, Canada, Oct. 2018, pp. 10–11.

[57] M. Askari and F. Keynia, "Mid-term electricity load forecasting by a new composite method based on optimal learning MLP algorithm," *IET Gener., Transmiss. Distrib.*, vol. 14, no. 5, pp. 845–852, Mar. 2020.

YING-YI HONG (Senior Member, IEEE) received the B.S.E.E. degree from Chung Yuan Christian University (CYCU), Taiwan, in 1984, the M.S.E.E. degree from National Cheng Kung University (NCKU), Taiwan, in 1986, and the Ph.D. degree from the Department of Electrical Engineering from National Tsing-Hua University (NTHU), Taiwan, in December 1990.

Sponsored by the Ministry of Education, China, he conducted Research with the Department of Electrical Engineering, University of Washington, Seattle, from August 1989 to August 1990. He has been with CYCU, since 1991. He was the Dean of the College of Electrical Engineering and Computer Science, CYCU, from 2006 to 2012. He was promoted to be a Distinguished Professor, due to his exceptional performance in research, leadership, teamwork, and international collaboration, in 2012. From 2012 to 2018, he was a Secretary General with CYCU, where he is currently the Dean of the Research and Development. His research interests include power system analysis and artificial intelligence applications. He received the Outstanding Professor of Electrical Engineering Award from the Chinese Institute of Electrical Engineering (CIEE), Taiwan, in 2006. He was the Chair of the IEEE PES Taipei Chapter, in 2001.



JONATHAN V. TAYLAR received the B.Sc. degree in computer engineering from the AMA Computer College, East Rizal, Antipolo, in 2006, and the M.Sc. degree in computer science from AMA University, Quezon City, Philippines, in 2008. He is currently pursuing the Doctor of Engineering degree in computer engineering with the Technological Institute of the Philippines, Quezon City.

Since 2015, he has been an Assistant Professor with the Computer Engineering Department, Technological Institute of the Philippines. He is currently conducting Technical Research with the Department of Electrical Engineering, Chung Yuan Christian University, Taoyuan, Taiwan. His research interests include machine learning, artificial intelligence, and deep learning.

ARNEL C. FAJARDO (Member, IEEE) received the B.Sc. degree in electrical engineering from the Mapúa Institute of Technology, Manila, in 1991, the M.Sc. degree in computer science from La Salle University, Manila, in 1999, and the Ph.D. degree in computer engineering from Hanbat National University, Daejon, South Korea, in 2014.

From 1999 to 2010, he was a lecturer with various universities, Philippines. He is currently a Professorial Lecturer with the Technological Institute of the Philippines, Quezon City (TIPQC), Philippines, and a Program Evaluator/Accreditor in information technology and computer science with the Philippine Accrediting Association of Schools, Colleges and Universities (PAASCU). He is also the Head of the Research and Development and a Senior Assistant Vice President with Manuel L. Quezon University, Quezon City. He is the author of more than 20 journal articles and conference proceedings. His research interests include speech recognition, artificial intelligence, and engineering education.