Article

Equation Based New Methods for Residential Load Forecasting

S. M. Mahfuz Alam and Mohd. Hasan Ali *

Department of EECE, The University of Memphis, Memphis, TN 38152, USA; smalam@memphis.edu
* Correspondence: mhi@memphis.edu

Received: 23 October 2020; Accepted: 27 November 2020; Published: 2 December 2020

Abstract: This work proposes two non-linear and one linear equation-based system for residential load forecasting considering heating degree days, cooling degree days, occupancy, and day type, which are applicable to any residential building with small sets of smart meter data. The coefficients of the proposed nonlinear and linear equations are tuned by particle swarm optimization (PSO) and the multiple linear regression method, respectively. For the purpose of comparison, a subtractive clustering based adaptive neuro fuzzy inference system (ANFIS), random forests, gradient boosting trees, and long-term short memory neural network, conventional and modified support vector regression methods were considered. Simulations have been performed in MATLAB environment, and all the methods were tested with randomly chosen 30 days data of a residential building in Memphis City for energy consumption prediction. The absolute average error, root mean square error, and mean average percentage errors are tabulated and considered as performance indices. The efficacy of the proposed systems for residential load forecasting over the other systems have been validated by both simulation results and performance indices, which indicate that the proposed equation-based systems have the lowest absolute average errors, root mean square errors, and mean average percentage errors compared to the other methods. In addition, the proposed systems can be easily practically implemented.

Keywords: adaptive neuro fuzzy inference system (ANFIS); random forest (RF); gradient boosting trees; long term short memory (LSTM); equation-based prediction system; load forecasting; smart buildings

1. Introduction

The energy utilization in residential and commercial buildings all over the USA is almost 40% of the overall energy generation. With the increase of luxury requirement of residents, the energy consumption is ever-increasing [1,2]. Therefore, providing the required power by grid is a hard task, especially during peak hours of the days. However, this problem can be solved in two ways. Firstly, by proper planning and allocation of energy resources by the grid, adequate power can be supplied to the consumers. Secondly, by implementing effective demand-side energy management system in the smart building that is capable of scheduling the load efficiently, the total cost of energy can be reduced by utilizing less loads that are operated by the grid power during the peak hours without affecting the consumers’ comfort demands [3,4]. An efficient load forecasting system helps the buildings’ energy management system schedule the loads ahead of time, operate the energy sources and energy storage systems effectively during peak hours to reduce the cost of energy and remove burden on the grids [5–7]. It also creates possibility for the smart building to sell energy to the grid during peak hours to achieve some incentives [8]. Moreover, with the knowledge of load forecasting, the grid can allocate
resources ahead of time and efficiently to meet up with the load demands [9,10]. Therefore, researchers have been investigating on improved and effective load forecasting methods over the last two decades.

Based on the forecast horizon or time scale, the load forecasting is classified generally into three categories, namely short-term load forecasting (STLF), medium-term load forecasting (MTLF), and long-term load forecasting (LTLF) [11,12]. Moreover, among various types of methods that are found in the literature, the most common are either time series or regression model type. The performance of time series models such as exponential smoothing, autoregressive integrated moving average (ARIMA) model, etc., depend upon the correlation between the loads and their previous values and availability of very large data set [11,13–15]. Another popular conventional method that is found in the literature is regression trees [16,17]. The random forest is a homogeneous ensemble approach with the combination of many decision trees without dependency on each other [18]. The drawback of the random forest method is its inability to extrapolate meaning; its prediction range is confined by the range in the training data as it takes the average value of all the trees. Moreover, it can be overfitted if the data set is large or noisy. Gradient boosting tree is another ensemble method that has been used for prediction [19,20]. The contrast between gradient boosting tree and random forest is that gradient boosting utilizes one tree for error minimization based on the experience of the previous tree. However, the gradient boosting tree method is more vulnerable to be overfitted in the presence of noise during data training and more parameters are needed to be tuned as compared to random forest. However, once it is properly tuned, it performs better than the random forest approach. Moreover, multiple linear regression-based forecasting method is found in the literature. The drawback of this approach is that it performs well for linear systems only, whereas the buildings’ loads are mostly non-linear in nature and the power consumption is non-linear as well [21,22]. Therefore, researchers have been focusing more on artificial intelligent based prediction system because of its ability to predict non-linear loads well under different indoor and outdoor conditions [23].

The artificial intelligence-based methods include fuzzy logic (FL), adaptive neuro fuzzy inference system (ANFIS), artificial neural network (ANN), support vector machine (SVM), etc. Among the artificial intelligence systems, the ANN method is found to be popular for load forecasting [24–28]. Moreover, a multi block neural network with a view to predicting price and load has been proposed in a recent work [29]. In addition, the Ridgelet and Elman neural networks-based load forecasting has been described in another work [30]. However, the ANN method requires a lot of historical data during training and validation stage for future data prediction [11]. In addition to that, the performance of the ANN depends upon several factors such as correlation between the inputs and output, the proper and efficient tuning of weight and bias of the hidden and output layer [24].

Therefore, in order to get better prediction method, the authors proposed a new two input fuzzy logic system for residential load forecasting that performs better than the ANN system [31]. Between the two inputs of the fuzzy systems, one is the temperature, and the other is a variable that is calculated from the occupancy number and day type. In another work, the fuzzy based peak energy management system is proposed for the industrial consumer [32]. It is to note here that the fuzzy logic is a non-linear system that operates on IF-THEN logic [33]. In addition, a fuzzy system is a slow system as it operates on the fuzzy rules that depend on the number of inputs and the membership function for each input. If each input has m membership functions and there are n inputs in the fuzzy system, then m^n rules need to be evaluated for each iteration of the fuzzy system, and therefore it is practically not suitable to implement, especially if the number of inputs exceeds two as the system becomes slower. Moreover, a new subtractive clustering based ANFIS system is proposed by authors, where the temperature and another variable being calculated from occupancy and day type, are considered as inputs, and the proposed ANFIS system performed better than the conventional ANN system [34]. The ANFIS system is a combination of both neural network and fuzzy system. Therefore, the ANFIS system requires a lot of data for the training and validation of the system than is required for neural network. In addition, similar to the fuzzy system, the ANFIS system becomes a slower system to be implemented practically for prediction if the number of inputs exceeds two.
Moreover, in recent times, the upgraded version of the recurrent neural network, named the long short term memory (LSTM) model, has been popular for forecasting [35–37]. The LSTM operates well where the conventional recurrent network fails with a large scale of sequential input data. However, the LSTM is more complicated than conventional neural networks, and, being a black box, it lacks interpretability. In addition, it does not perform well in case of small input data, if the parameters are not properly tuned or input data are not sequential.

In addition, stochastic optimization-based prediction systems have been gaining popularity for forecasting. They are utilized mainly in the case of uncertainty in the system. A hybrid stochastic approach, for bidding strategies and demand uncertainties of large consumer, is proposed in a work [38]. Similarly, stochastic optimizations have been proposed for dealing with uncertainties in cooling demand [39] and for risk assessments of large consumer group [40].

However, the energy consumption of a residential building depends upon the habits of residents living there, responses to different environmental conditions, mode of comfort, etc. Although the consumers’ reference comfort temperature during the different conditions mentioned above can be different, however, in general, it is assumed that in USA, if the outside average temperature of the day is 65 °F, then no heating and cooling are required to be comfortable [41]. Therefore, if the energy consumption is categorized based on heating degree days (HDD), cooling degree days (CDD), number of occupants, and the day type, the uncertainty of energy reduces a lot. HDD is a term that is used for showing how much the day’s temperature is below the consumers’ reference comfort temperature (65 °F) and this constant temperature is used for HDD calculation all over USA for all the seasons [42,43]. Similarly, CDD is a term that defines how much the temperature is above the consumers’ reference comfort temperature (65 °F) [42,43].

Based on the above background, a new method, that is practically implementable and does not need a lot of data for training but works better than conventional prediction systems, requires attention and investigation. Therefore, having been motivated by this fact, this paper proposes a new method based on both non-linear and linear equations for residential load forecasting. The coefficients in the proposed non-linear equations have been tuned by the Particle Swarm optimization (PSO) algorithm. The PSO is a stochastic optimization technique that has advantageous inherent features such as a fast convergence rate as compared to other optimization techniques such as the genetic algorithm and provide more effective solutions. Moreover, it is practically implementable and has been applied in different applications [44–48]. The coefficients of the proposed linear equation-based system are tuned by multiple linear regression (MLR) using the least squares approach, which is available in MATLAB software.

The main contributions of this paper are summarized as follows:

- Three generalized equations are developed for predicting load consumption based on the HDD, CDD, occupancy, and day type. The coefficients of the non-linear equations and linear equation are optimized by the well-known PSO and multiple linear regression method, respectively.
- In order to see efficacy of the proposed equation-based methods, in predicting the loads, their performance have been compared with that of a recently published forecasting method such as the subtractive clustering based ANFIS approach, random forest, gradient boosting trees and LSTM, and conventional and modified support vector regression models.

In this work, the predicted data for all methods are simulated in MATLAB software and different errors are considered as performance indices to validate the efficacy of the proposed equations-based prediction systems.

The rest of the paper is organized as follows. In Section 2, the proposed equation-based prediction systems are described. Section 3 explains the conventional forecasting method, i.e., the ANFIS system, random forest, gradient boosting, LSTM, conventional and modified support vector regression. Simulation results are presented and explained in Section 4. The conclusion and future research directions are provided in Section 5. Finally, the references are enlisted.
2. Proposed Equation Based Prediction Methods

The load consumption of a building depends highly on temperature. The increase in temperature increases the load consumption if the temperature is above a certain temperature, which in general is 65 °F in USA, due to a higher cooling requirement. In addition, if the temperature goes below the same temperature mentioned above, the load consumption increases due to a higher requirement of heating. Therefore, energy consumption of a residential building is dependent upon HDD and CDD, which represent temperatures below or above 65 °F. Based on this fact, the energy consumption, \( e \), can be expressed as the following:

\[
\begin{align*}
    e & \propto HDD, \\
    e & \propto CDD.
\end{align*}
\]

Moreover, for the same temperature, HDD, CDD, the energy consumption increases with an increase in the number of occupants and vice versa in the same apartment. Therefore,

\[
    e \propto \text{Occupant}.
\]

In addition, the energy consumption pattern of a building is different for a normal working day, weekend, any special day, or special occasion. The special day depends on the family living in a building when there may have some religious festival celebrations, some family events happening, or more than usual family members staying in the building for some reasons. In addition, it can be a normal working day or weekend or even holiday. Therefore,

\[
    e \propto \text{Day type}.
\]

Therefore, based on the above discussion, three types of equations, as shown in (1) to (3) below, have been developed for load predictions in this work. The first equation is linear in nature as variable HDDC, occupant number \( (O) \), and day type \( (D) \), and values are linearly multiplied with the coefficients to predict the total energy consumption of the day. Moreover, the other two equations are non-linear in nature as some power values of HDDC, \( O \) are multiplied with the coefficients, whereas \( D \) is used as power for Equations (2) and (3). The exponential component is used in (2), whereas the variable is a constant whose values are determined by the optimization algorithm for (3).

\[
\begin{align*}
    e &= C_1 \text{HDDC} + C_2 O + C_3 D + C_4, \\
    e &= C_5 \text{HDDC}^n + C_6 O^n + C_7 \exp^D + C_8, \\
    e &= C_9 \text{HDDC}^p + C_{10} O^q + C_{11} D + C_{12},
\end{align*}
\]

where, \( e \) and \( O \) represent the total load consumptions in kWh in a day and number of occupants present on that day, respectively. HDDC represents the HDD values, which is the difference between the average day’s temperature and 65 °F if the temperature is below or equal to 65 °F. Moreover, HDDC represents the CDD values, which is the difference between 65 °F and the day’s average temperature values if the temperature is above 65 °F. The coefficients \( C_1, C_5, \) and \( C_9 \) depend on HDD or CDD values for (1) to (3), respectively. \( C_2, C_6, C_{10} \) are the coefficients for number of occupants. The coefficients \( C_3, C_7, \) and \( C_{11} \) vary with the day type. The coefficients \( C_4, C_8, \) and \( C_{12} \) are considered to be off-sets that are dependent on HDD, CDD, occupancy and the day-type. The values for \( D \) for normal working days, weekend, and special days are considered to be 0, 1, and 2, respectively, for this work. The equations proposed in (1) and (2) to (3) are linear and non-linear in nature, respectively, whose performance certainly depends on the properly tuned values of the coefficients with the varying HDD, CDD values, occupancy, or the type of the days. Therefore, multiple linear regression method and PSO algorithm have been utilized to obtain the coefficients of the linear equation in (1) and non-linear equations
proposed in (2) and (3), respectively, in order to predict the optimal total energy consumption of the day.

The working principle of the equation-based methods for residential load predictions, which consider the HDD, CDD, occupancy and the values of D based on normal working days, weekends, or special days as inputs (x), are shown in Figure 1. In this work, generalized equations are formulated based on the inputs. The dotted line portion shown in Figure 1 represents the equation-based prediction systems. First the inputs (x) are fed into the equation-based prediction system so that the ranges of the input variables are selected. Once the ranges of the variables are selected, the MLR/PSO tuned coefficient values are sent to the main equation block where Equations (1) to (3) are utilized to predict the energy consumption based on the inputs and the coefficients. The multiple linear regression method (MLR) or the PSO method provides the optimized coefficient (C₁ … C₉/C₁₀ … C₁₂) values for the proposed Equations (1)–(3) based on the range of inputs (i.e., HDD, CDD, O, D) which are summarized in Tables 1–3. These optimized tuned coefficients are obtained from the previous input data and the energy consumption data obtained from the smart meter.

Table 1. Coefficients of Equation (1) determined by the MLR method.

<table>
<thead>
<tr>
<th>Condition</th>
<th>C₁</th>
<th>C₂</th>
<th>C₃</th>
<th>C₄</th>
<th>C₅</th>
<th>C₆</th>
<th>C₇</th>
<th>C₈</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDD ≥ 17, O ≥ 0, D ≥ 0</td>
<td>0.882</td>
<td>1.499</td>
<td>10.260</td>
<td>0.00</td>
<td>1.662</td>
<td>6.779</td>
<td>8.723</td>
<td>-4.370</td>
</tr>
<tr>
<td>0 ≤ CDD &lt; 17, O ≥ 0, D ≥ 0</td>
<td>-1.270</td>
<td>11.732</td>
<td>5.839</td>
<td>-10.9</td>
<td>12.784</td>
<td>4.393</td>
<td>-17.94</td>
<td>7.641</td>
</tr>
<tr>
<td>0 ≤ HDD &lt; 20, O ≥ 0, D ≥ 0</td>
<td>-10.59</td>
<td>13.146</td>
<td>6.80</td>
<td>10.9</td>
<td>12.784</td>
<td>4.393</td>
<td>-17.94</td>
<td>7.641</td>
</tr>
</tbody>
</table>

Table 2. Coefficients of Equation (2) determined by the PSO method.

<table>
<thead>
<tr>
<th>Condition</th>
<th>m</th>
<th>n</th>
<th>C₅</th>
<th>C₆</th>
<th>C₇</th>
<th>C₈</th>
<th>C₉</th>
<th>C₁₀</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDD ≥ 17, O ≥ 2, D ≥ 1</td>
<td>0.001</td>
<td>1.356</td>
<td>8.487</td>
<td>10.121</td>
<td>2.624</td>
<td>-8.716</td>
<td>0.001</td>
<td>0.695</td>
</tr>
<tr>
<td>CDD ≥ 17, O ≥ 2, D = 0</td>
<td>0.462</td>
<td>0.389</td>
<td>14.455</td>
<td>11.955</td>
<td>5.289</td>
<td>-1.656</td>
<td>0.001</td>
<td>0.010</td>
</tr>
<tr>
<td>10 ≤ CDD &lt; 17, O ≥ 1, D ≤ 1</td>
<td>0.001</td>
<td>1.356</td>
<td>8.487</td>
<td>10.121</td>
<td>2.624</td>
<td>-8.716</td>
<td>0.001</td>
<td>0.695</td>
</tr>
<tr>
<td>3 ≤ CDD ≤ 9, O ≥ 3, D = 2</td>
<td>0.368</td>
<td>0.974</td>
<td>2.964</td>
<td>6.079</td>
<td>2.269</td>
<td>-9.368</td>
<td>0.001</td>
<td>0.462</td>
</tr>
<tr>
<td>0 ≤ HDD &lt; 30, O ≥ 0, D &gt; 0</td>
<td>0.253</td>
<td>0.852</td>
<td>8.723</td>
<td>14.206</td>
<td>0.010</td>
<td>-18.75</td>
<td>0.001</td>
<td>0.695</td>
</tr>
<tr>
<td>HDD ≥ 30, O ≥ 0, D &gt; 0</td>
<td>0.0291</td>
<td>1.2687</td>
<td>0.0100</td>
<td>3.8381</td>
<td>0.0100</td>
<td>3.0354</td>
<td>0.001</td>
<td>0.695</td>
</tr>
</tbody>
</table>

Table 3. Coefficients of Equation (3) determined by the PSO method.

<table>
<thead>
<tr>
<th>Condition</th>
<th>p</th>
<th>q</th>
<th>a</th>
<th>C₉</th>
<th>C₁₀</th>
<th>C₁₁</th>
<th>C₁₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDD ≥ 17, O ≥ 2, D ≥ 1</td>
<td>0.952</td>
<td>1.056</td>
<td>-6.80</td>
<td>3.070</td>
<td>13.146</td>
<td>-3.545</td>
<td>-6.80</td>
</tr>
<tr>
<td>HDD ≥ 17, O ≥ 2, D ≥ 0</td>
<td>0.212</td>
<td>0.989</td>
<td>-10.9</td>
<td>12.784</td>
<td>4.393</td>
<td>-17.94</td>
<td>-10.9</td>
</tr>
<tr>
<td>10 ≤ CDD ≤ 16, O ≤ 1, D ≤ 1</td>
<td>0.001</td>
<td>1.375</td>
<td>1.979</td>
<td>12.598</td>
<td>6.295</td>
<td>-10.99</td>
<td>1.979</td>
</tr>
<tr>
<td>3 ≤ CDD ≤ 9, O ≥ 3, D = 2</td>
<td>0.001</td>
<td>0.389</td>
<td>0.574</td>
<td>0.010</td>
<td>13.474</td>
<td>20.000</td>
<td>0.574</td>
</tr>
<tr>
<td>0 ≤ HDD &lt; 30, O ≥ 0, D ≥ 0</td>
<td>0.728</td>
<td>1.100</td>
<td>0.407</td>
<td>0.551</td>
<td>9.557</td>
<td>-5.325</td>
<td>0.407</td>
</tr>
<tr>
<td>HDD ≥ 30, O ≥ 0, D ≥ 0</td>
<td>0.0885</td>
<td>1.115</td>
<td>1.330</td>
<td>1.736</td>
<td>5.021</td>
<td>-15.914</td>
<td>1.330</td>
</tr>
<tr>
<td>0.01 ≤ HDD &lt; 10, O ≥ 0, D ≥ 0</td>
<td>0.368</td>
<td>1.137</td>
<td>0.933</td>
<td>6.257</td>
<td>6.409</td>
<td>5.787</td>
<td>0.933</td>
</tr>
<tr>
<td>0.01 ≤ HDD &lt; 10, O ≥ 0, D ≥ 0</td>
<td>0.180</td>
<td>1.143</td>
<td>-1.270</td>
<td>11.732</td>
<td>5.839</td>
<td>-10.59</td>
<td>-1.270</td>
</tr>
</tbody>
</table>
2.1. Parameter Tuning by Multiple Linear Regression (MLR) Algorithm

In MATLAB, the command, regress is used for calculating the coefficients of the linear model, which has the following format:

\[ e = Cx, \]  \hspace{1cm} (4)

subject to

\[ \sum (y - e)^2 = \text{minimum}, \]

where the input matrix, \( x = [HCDD; O; D; U] \) \( C = [C_1 \ C_2 \ C_3 \ C_4] \) and \( y \) represent the anticipated output obtained from the smart meter. \( U \) is a unity vector of length of HDCC vector to determine the values of \( C_4 \) by the multiple linear regression algorithm and introduced in the \( x \) matrix as dummy as for each set of data, the columns number of \( C \) matrix should be equal to the rows number of \( x \) matrix. By matrix multiplication of \( C \) and \( x \) matrix, the predicted output (\( e \)) is calculated and put to the condition shown above until the coefficient values (\( C_1, \ldots, C_4 \)), for which the summation of square of the difference between the anticipated output (\( y \)) and predicted output (\( e \)) gets minimum.

2.2. Parameter Tuning by Particle Swarm Optimization (PSO) Algorithm

As already mentioned, in this work, the PSO method has been used for parameters tuning of the non-linear equations shown in (2) and (3). It has been widely applied in applications such as energy management [44,45], load predictions [46–48], etc. It is very easy to implement and has faster convergence speed and effective over other optimization algorithms such as the genetic algorithm [45].

In PSO, a random number of particles are chosen for search space and the objective function is defined. Based on the cost function at any current location, the optimal position and cost are determined and updated among the particles. Each particle then finds its new position based on its current position, previous velocity and global optimal location among the particles. After updating its positions and velocity vectors, again the best position and cost among the particles are circulated and updated. Therefore, by updating the situations (position and velocity vectors) and collaborating the information of optimal best location and optimal cost, the swarm as a group reaches its optimal goal.

The PSO algorithm is characterized by the two-model equations of velocity and position vector in an N-dimensional solution space as shown below:

\[ v_{i}^{k+1} = w v_{i}^{k} + c_1 r_1 (p_{i}^{k} - x_{i}^{k}) + c_2 r_2 (p_{g}^{k} - x_{i}^{k}), \]  \hspace{1cm} (5)

\[ x_{i}^{k+1} = x_{i}^{k} + v_{i}^{k+1}, \]  \hspace{1cm} (6)

where \( v_{i}^{k+1} \) represents \( i^{th} \) particle velocity of \( (k + 1)^{th} \) iteration of N dimensional search space. Similarly, \( x_{i}^{k} \) corresponds to \( i^{th} \) particle velocity of \( k^{th} \) iteration. \( p_{i}^{k} \) and \( p_{g}^{k} \) correspond to the individual best position of the \( i \) particle and global best position of the swarm, respectively. Moreover, \( r_1 \) and \( r_2 \) are randomly chosen numbers, which are uniformly distributed between 0 and 1. \( c_1 \) and \( c_2 \) are known as learning factors which control the significance of the best solution. The values for both learning factors are chosen to be 2. The value for the inertia coefficient, \( w \) for each iteration number is calculated using the following equation:

\[ w = w_{\text{max}} - \frac{t (w_{\text{max}} - w_{\text{min}})}{\text{MaxI}} , \]  \hspace{1cm} (7)

where, \( w_{\text{max}} \) and \( w_{\text{min}} \) represent the upper and lower value of \( w \) and \( t \), respectively, \( \text{MaxI} \) correspond to the current iteration number and maximum iteration number, respectively.

The objective function for the current work is considered as follows:

\[ f_{\text{obj}} = y - e , \]  \hspace{1cm} (8)
subject to
\[ |y - e| = \text{minimum}. \]

The procedure of the PSO algorithm is described as follows:

- Initialization:
  1. Load the input \((x)\) and anticipated output \((y)\) value based on the smart meter data.
  2. Set the parameters of the PSO obtained from several trials which gave the optimal output.

  - Search space dimension = 1
  - Population size = 30
  - Maximum number of iterations = 150
  - \(w_{\text{max}} = 0.9\) and \(w_{\text{min}} = 0.2\)
  - Penalty factor = 500

- Iteration:
  1. Randomly generation of velocity and position vectors, which is done by PSO.
  2. Evaluate the cost function based on (5) to measure the fitness values for the corresponding inputs.
  3. Start the iteration
  - Run the algorithm 150 times.
  - Based on (5) and (6), update the velocity and position vectors.
  - Determine the predicted energy \((e)\) for the predicted horizon. If the constraint is violated, then add the penalty factor.
  - Determine the cost function.
  - Update the individual best and global best values based on the cost function.
  - Update the inertia weight.
  - Repeat step 3 until maximum number of iterations is reached.

After the optimal coefficients are obtained from the MLR and PSO, the coefficients are put into (1) to (3) to get the predicted outputs. The coefficients, based on different HDD, CDD, occupancy and day type condition, as determined by the MLR and PSO methods, are shown in Tables 1–3, respectively.

Interpretability is the main advantage of this proposed method. The model explains the energy consumption based on the heating degree days (HDD), cooling degree days (CDD), occupancy, and the day type. The proposed equation-based system is practically implementable as it needs only three parameters (temperature, number of occupants, type of the day.). The predicted temperature information for future days can be easily found online. The number of occupants can be inserted by the consumer, or a motion detector can be placed inside the building to count the number of occupants. Moreover, normal working days and weekend information can be available from an online calendar and the special day information can be inserted by the consumers. Once the coefficients and the temperature range are known to consumers, they even can calculate the energy consumption by hands. Moreover, it requires moderate amount of data (energy consumption, HDD, CDD, occupancy, day type) for parameter coefficient tuning by MLR and PSO. It is very convenient for practical implementation. However, the energy consumption of a residential building depends upon the habits of residents living there, responses to different environmental condition chance, mode of comfort (the usage of appliances based on consumer comfort desire under different conditions), etc. Therefore, these three equations can be implemented for any building provided that the coefficient is re-tuned based on the energy consumption pattern and other conditions such as country, region, and location.

In the first condition in Table 1, it refers to the temperatures for which CDD will be 17 °F above the reference temperature (65 °F). All temperatures equal to or higher than 82 °F (65 °F + 17 °F), would have an equivalent value for CDD of 17 °F or higher. Similarly, in the second condition, the values of CDD between 0 to less than 17 °F refer to all the values of temperatures from 65 °F to 81 °F (below 82 °F). Moreover, the value of HDD in the third condition refers to all the temperature less than or equal to 20 °F or higher than the reference value 65 °F. In this case, all the temperatures that will be in the range 0 °F to 45 °F (65 °F – 20 °F) will be equivalent value for HDD of 20 °F or higher. Finally, all the temperatures in the range above 45 °F (65 °F – 20 °F) to 64.9 °F (65 °F – 0.1 °F) would be equivalent for HDD to have
value less than 20 °F to 0.1 °F value. Therefore, by choosing these four ranges, all temperatures are considered. Similarly, the temperature of different ranges in terms of HDD and CDD are considered in Tables 2 and 3.

It is important to note that the HDD and CDD values are calculated based on the constant reference temperature (65 °F) for USA. However, the consumers’ temperature comfort for different seasons and conditions can be different. Therefore, in order to cope with both conditions and predict the accurate energy consumption with HDD/CDD, the coefficients (C1 for Equation (1), C5 for Equation (2), and C9 for Equation (3)) are tuned and based on HDD/CDD values for the defined range of HDD/CDD, and represent the energy variation with per degree variation of HDD/CDD (kWh/°F). Moreover, if the above methods are used for other residential places located in others countries, regions, etc., then the HDD/CDD values should be calculated based on that region’s reference temperature and the coefficient should be tuned accordingly.

3. Conventional Methods

As already mentioned, in this work, the performance of the proposed equation-based methods has been compared with that of the conventional methods such as the ANFIS, random forest, gradient boosting trees, and LSTM. These conventional methods are described below.

3.1. Adaptive Neuro Fuzzy Inference System (ANFIS) Based Load Forecasting

The ANFIS is an intelligent model with the inherent contribution of both a neural network and a fuzzy system. In this work, a Sugeno-type ANFIS system is considered. The ANFIS system is governed by two major stages, namely antecedent and conclusion. Both parts are related to each other by fuzzy rules. For the chosen Sugeno type ANFIS system, the fuzzy rules are formulated by the following equation [34]:

\[
If \ (x_1 = A_i) \ and \ (x_2 = B_i) \ then \ f_i = p_i x_1 + q_i x_2 + r_i
\]

where, \(x_1\) and \(x_2\) correspond to the inputs to the ANFIS system. Two inputs that have been chosen, are temperature \((x_1)\) and a variable, \(R\) \((x_2)\), as shown in (10). \(A_i\) and \(B_i\) represent the fuzzy sets. Therefore, \(f_i\) indicates the output that is governed by the fuzzy rules. For example, temperature corresponds to \(A_1\) and \(R\) value corresponds to \(B_1\), rule 1 of the output would be: \(f_1 = p_1 A_1 + q_1 B_1 + r_1\). During the training process, the parameters (i.e., \(p_i\), \(q_i\), and \(r_i\)) are calculated. The input, \(R\) is determined by (10):

\[
R = \text{occupant} + 1.5 \times d.
\]

The value of \(d\) can be 0, 1, and 2 based on normal working days, weekend, and special days, respectively. Therefore, if the number of occupants for a day is 5, and the day is a normal working day \((d = 0)\), the value of \(R\) would be 5. If the day is a weekend \((d = 1)\) or special day \((d = 2)\), for the same number of occupants \((5)\), the values of \(R\) would be 6.5 and 8, respectively.

In the ANFIS system, at first the data is utilized during the training process and the rules are extracted and membership functions types and their positions are determined through training and testing. Finally, the results are used for future predictions. For this work, during training, temperature, \(R\) values, and output energy consumption data of 304 days are provided. The parameters for the input (temperature, \(R\)) and output (total energy consumption) membership functions are tuned by the hybrid algorithm that utilizes the backpropagation method for the parameter of input membership function. In addition, output membership function parameters are optimized by the least square estimation method. Subtractive clustering defines the number of the fuzzy rules along with the number of membership functions and membership type. Therefore, the subtractive method is very useful if the data pattern is unknown, as well as if one is unsure as whether or not to choose the number of membership function with the membership type and center position.
The parameters of subtractive clustering are chosen from [34]. In normal fuzzy system, if both inputs have 10 membership functions, then the total fuzzy rules would have been 100, which have to be analyzed for each input data. However, for the chosen subtractive clustering parameters, each input has 10 membership functions and the total number of fuzzy rules is 10, as shown in Figure 2, which makes the subtractive clustering beneficial and the system faster. The minimum error and number of epochs are chosen to be 0 and 500, respectively. The minimal root-mean-square error is found to be 5.13 after 500 epochs. The tuned Gaussian fuzzy membership functions are shown in Figure 3. The parameters of ANFIS system are used from [34].

**Figure 2.** ANFIS structure. (It is a system-built figure, only blue dots are present as only “And” rules are used).

**Figure 3.** Input membership functions (a) input 1 (b) input 2.
3.2. Random Forest Based Load Forecasting

Random forest is an ensemble approach that emphasizes the predictions of all the decision trees that are independent upon each other [49]. The sample size is randomly selected and fitted into a regression tree. The process is known as bagging and the selected sample is called bootstrap. This sample is replaced with another random sample each time. The probability of all the observations is assumed to be same. The bagging algorithm then implements the classification and regression tree (CART) algorithm to obtain a set of regression trees and finally averages the output of all trees based on the following equation:

\[
\hat{Y'} = \frac{1}{r} \sum_{i=1}^{r} \hat{h}(X', S_n^{\theta_i}),
\]

where, \(\hat{Y'}\) is the output estimation based on new input \(X'\) and \(\hat{h}(X', S_n^{\theta_i})\) is the predicted output of bootstrap sample of \(S_n\). \(\theta_i\) represents a randomly chosen variable having identical distribution.

For this method, the input variables considered are temperature, occupancy, and day type. The energy consumption per day is the output of the prediction system. The unbiased importance of input variables that are measured using the out of bag method and the number of levels, is shown in Figure 4.

![Random forest predictors' importance and number of levels](image)

Figure 4. Random forest predictors’ (a) importance, and (b) number of levels.

The parameter of this method, optimized by the Bayesian optimization algorithm [50], are summarized in the Table 4.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Number of Split</td>
<td>55</td>
</tr>
<tr>
<td>Minimum Leaf Size</td>
<td>6</td>
</tr>
<tr>
<td>Number of Learning Cycles</td>
<td>494</td>
</tr>
</tbody>
</table>

3.3. Gradient Boosting Trees Based Load Forecasting

The gradient boosting is an additive model that is characterized by the following equation [51]:

\[
F_m(x) = F_{m-1}(x) + h_m(x),
\]

where \(F_m(x)\) represents the prediction sum of all \(m\) regression trees and \(h_m(x)\) is the fixed sized regression trees. In MATLAB, the least square boosting (LSBoost) is used for regression [52,53]. At each iteration, the ensemble adds a new tree to the difference between the response observed and the summation of prediction of all trees used before. The LSBoost is efficient in minimizing the mean-squared error. Similar to the random forest method, the variables such as temperature, occupancy, and day type...
are considered as inputs for this method. The energy consumption per day is the output of the prediction system. The parameters of this method, optimized by the Bayesian optimization algorithm, are summarized in Table 5.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Number of Split</td>
<td>2</td>
</tr>
<tr>
<td>Minimum Leaf Size</td>
<td>1</td>
</tr>
<tr>
<td>Number of Learning Cycles</td>
<td>395</td>
</tr>
<tr>
<td>Learn Rate</td>
<td>0.043934</td>
</tr>
</tbody>
</table>

### 3.4. LSTM Based Load Forecasting

The LSTM is an improved version recurrent neural network (RNN) with added cell state and gates and thus it has the ability to overcome the gradient vanishing problem that the conventional RNN has [35,36]. The LSTM is characterized by the following sets of equations:

\[
\begin{align*}
    f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f), \\
    i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i), \\
    o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o), \\
    \tilde{C}_t &= \tanh(W_C[h_{t-1}, x_t] + b_C), \\
    C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \\
    h_t &= o_t \odot \tanh(C_t),
\end{align*}
\]

where, \(f_t\) represents forget gates that control the amount of previous states to be reflected on the current states. It is the input and \(o_t\) is the output gates that decide the amount of new information to update the cell state and to output depending on cell state. \(\sigma\) keeps the output values between 0 to 1. All the gates are updated based on current input \(x_t\) and previous output \(h_{t-1}\). \(C_t\) and \(\tilde{C}_t\) represent cell state and the value required for calculating cell state, respectively. For the LSTM based load forecasting, the input variables are temperature, occupancy, and day type. The training of the LSTM approach is shown in Figure 5. For the LSTM model parameters, the Adam optimization approach is used [34] and the parameters for LSTM are shown in Table 6.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Hidden Units</td>
<td>500</td>
</tr>
<tr>
<td>Fully Connected Layer</td>
<td>150</td>
</tr>
<tr>
<td>Dropout Layer</td>
<td>0.1</td>
</tr>
<tr>
<td>Maximum Number of Epoch</td>
<td>250</td>
</tr>
<tr>
<td>Minimum Batch Size</td>
<td>3</td>
</tr>
<tr>
<td>Initial Learn Rate</td>
<td>0.005</td>
</tr>
<tr>
<td>Learn Rate Drop Period</td>
<td>125</td>
</tr>
<tr>
<td>Learn Rate Drop Factor</td>
<td>0.2</td>
</tr>
</tbody>
</table>
The modified support vector regression (SVR)-based prediction method involves three stages for residential buildings energy consumption predictions, as shown in Figure 6. In the first stage, the previous historical data inputs ($x_{tr}$) and known energy consumptions ($y_{tr}$) are fed into the SVR training stage, which produce the values of $\beta_0, b_0$. $\beta_0$ has 14 values which correspond to coefficients for 14 input parameters such as temperature, humidity, wind speed, etc. The obtained values of $\beta_0, b_0$ by the SVR training system are then considered as the initial values for the PSO stage. In the PSO stage, the predicted inputs ($x$) and anticipated consumption ($y$), which can be obtained from smart meter by similar day/input approach, are inserted. As already mentioned, energy consumption in a residential building depends on the temperature range, other environmental conditions range, occupancy, or even the day type. Therefore, more sets of parameter values are required to be considered based on temperature range to predict the energy consumption more accurately. Therefore, four sets of $\beta_{optn}, b_{optn}$ values are generated by the PSO method based on the temperature range and one of four sets values of $\beta_{optn}, b_{optn}$ based on the corresponding temperature is used by the SVR equation to predict the energy consumption of the residential building, as shown in Figure 6, where $n = 1, 2, \ldots 4$.

**Figure 5.** LSTM model training.

### 3.5. Conventional and Modified Support Vector Regression Based Load Forecasting

The support vector regression, because of its dependence on kernel function, is considered as a nonparametric technique [54]. In MATLAB, epsilon-insensitive support vector regression is available.
in which the set of training data of both predictor variables \((x_{tr})\) and observed response values \((y_{tr})\) are provided with a view to deriving a function \(f(x)\) which will deviate from all \(y\) within the limit of \(\varepsilon\) values. Therefore, the equation for the \(f(x)\) can be expressed as shown in (19) [54,55].

\[
f(x) = x'\beta + b,
\]

where, \(x\) is the set of \(N\) observation, \(\beta\) and \(b\) represent the coefficients of input and bias, respectively. In order to formulate a convex optimization problem and to ensure that \(f(x)\) is as flat as possible, it is required to minimize the objective function, which can be represented by the following equation:

\[
J(\beta) = \frac{1}{2} \beta'\beta. \tag{20}
\]

Subject to

\[
\forall n : \left| y_n - (x'\beta + b) \right| \leq \varepsilon,
\]

where, \(\varepsilon\) is the residue. Since it might not be possible for \(f(x)\) to satisfy the constraint in (20) for all values of \(x\), two slack variables \(\xi_n\) and \(\xi_n^*\) are included with a view of maintaining the constraint shown in (21) for all values of \(x\). Therefore, the objective function presented in (20) can be rewritten as follows:

\[
J(\beta) = \frac{1}{2} \beta'\beta + C \sum_{n=1}^{N} (\xi_n + \xi_n^*). \tag{21}
\]

Which subjects to:

\[
\forall n : y_n - (x'\beta + b) \leq \varepsilon + \xi_n,
\]

\[
\forall n : (x'\beta + b) - y_n \leq \varepsilon + \xi_n^*,
\]

\[
\forall n : \xi_n \geq 0,
\]

\[
\forall n : \xi_n^* \geq 0,
\]

where, \(C\) is known as the box constraint that has the ability to control the penalty when the observation does not fall within the \(\varepsilon\) margin. It also controls the trades between the flatness of \(f(x)\) and maximum tolerable values beyond \(\varepsilon\) margin.

The linear \(\varepsilon\)-insensitive loss function can be expressed as:

\[
L_\varepsilon = \begin{cases} 0 & \text{if } |y - f(x)| \leq \varepsilon \\ |y - f(x)| & \text{otherwise} \end{cases},
\]

\[
L(\alpha) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (a_i - a_i^*)(x_j - a_i^*) G(x_i, x_j) + \varepsilon \sum_{i=1}^{N} (a_i + a_i^*) - \sum_{i=1}^{N} y_i(a_i - a_i^*). \tag{22}
\]

The non-linear support vector regression can be achieved using Lagrange dual formulations. Then, the objective function becomes as shown in (22). The constraints in (22) are:

\[
\sum_{n=1}^{N} (a_n - a_n^*) = 0,
\]

\[
\forall n : 0 \leq a_n \leq C,
\]

\[
\forall n : 0 \leq a_n^* \leq C,
\]

where, the linear Kernel function can be expressed as:

\[
G(x_i, x_j) = x_i'x_j. \tag{23}
\]
The objective function shown in (22) can be solved by the quadratic programming techniques. In this work, sequential minimal optimization method (SMO), which is a very popular approach for SVR problems, is considered. In SMO, a series of two-point optimization is considered and these two points are selected by a selection rule that is governed by second-order information. In SVR, the gradient vector is updated after each iteration by the following equation:

\[
(VL)_n = \sum_{i=1}^{N} \left( a_i - \alpha_i^* \right) G(x_i, x) + \epsilon - y_n, n \leq N \\
- \sum_{i=1}^{N} \left( a_i - \alpha_i^* \right) G(x_i, x) + \epsilon + y_n, n > N.
\] (24)

After the training process described in (19)–(24), the values of \( \beta_0, b_0 \) are obtained and then fed in the PSO stage for further optimizations. For PSO, all the methods and parameters are used the same, as described in Section 2.2.

After the optimal coefficients are obtained from the PSO based on the temperature range, input and anticipated output, the coefficients are put into (19) to get the predicted output.

Moreover, in this work, the conventional PSO tuned SVR method, as shown in Figure 7, has also been used. Likewise, the modified SVR system, the conventional system, also involves three stages for energy consumption predictions. The SVR training stage produces the \( \beta_0, b_0 \) for the PSO stage. Then, the PSO provides only one set of values of \( \beta_{opt}, b_{opt} \) based on the predicted inputs and anticipated consumption, which can be obtained from a smart meter using the similar day/input approach. Therefore, the SVR training system and the PSO stage are the same for both methods with the exception that the modified system considers the temperature range as an additional input. The coefficients, based on different temperatures for the modified SVR method and one set for all temperatures for the conventional SVR method are shown in Table 7, where all T values are in degree Fahrenheit (°F).

Figure 7. Block diagram of conventional PSO tuned support vector regression-based prediction system.

Table 7. Values of \( \beta_{opt}, b_{opt} \) Determined by the PSO Method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Conventional SVR</th>
<th>Modified SVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta_{opt} )</td>
<td>( \beta_{opt} )</td>
<td>( \beta_{opt} )</td>
</tr>
<tr>
<td>For all T</td>
<td>( T \geq 75 )</td>
<td>( 65 \leq T &lt; 75 )</td>
</tr>
<tr>
<td>-0.0050</td>
<td>-0.0050</td>
<td>-0.0050</td>
</tr>
<tr>
<td>0.0011</td>
<td>0.0011</td>
<td>0.0011</td>
</tr>
<tr>
<td>-0.0018</td>
<td>-0.0018</td>
<td>-0.0018</td>
</tr>
<tr>
<td>-0.0002</td>
<td>0.0349</td>
<td>0.0349</td>
</tr>
<tr>
<td>0.0256</td>
<td>0.0009</td>
<td>0.0009</td>
</tr>
<tr>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>0.0242</td>
<td>0.0242</td>
<td>0.0242</td>
</tr>
<tr>
<td>0.1662</td>
<td>0.1662</td>
<td>0.0008</td>
</tr>
<tr>
<td>7.2933</td>
<td>10.976</td>
<td>10.4529</td>
</tr>
<tr>
<td>2.3878</td>
<td>2.3878</td>
<td>2.3878</td>
</tr>
<tr>
<td>0.8940</td>
<td>0.6227</td>
<td>0.3571</td>
</tr>
<tr>
<td>0.3268</td>
<td>0.0074</td>
<td>1.4746</td>
</tr>
<tr>
<td>0.00003</td>
<td>0.00003</td>
<td>0.00003</td>
</tr>
<tr>
<td>0.0043</td>
<td>0.1285</td>
<td>0.0043</td>
</tr>
<tr>
<td>( b_{opt} )</td>
<td>( b_{opt} )</td>
<td>( b_{opt} )</td>
</tr>
<tr>
<td>-0.4182</td>
<td>-0.4182</td>
<td>-0.4182</td>
</tr>
</tbody>
</table>
4. Simulation Results and Discussion

4.1. Simulation Data and Conditions

In this work, the daily total energy demand and the average temperature data of the day were collected from an apartment located in 3571 Midland Avenue, Memphis, TN. The smart energy meter (meter 54BKW988882) data is available in the MLGW web account. Moreover, the number of occupants present at any day and type of the day information were collected from the residents in the building. A total of 334 days of data (334 sets of data) of average temperatures for a given day, average number of occupants for the day, day type, were collected. Moreover, out of these data, randomly chosen 30 days (30 sets of data) data were used for the prediction of total energy consumption per day for comparison purposes and rest 304 days data were used for the ANFIS, random forest, LSBoost, and LSTM network methods for their training and validation. Similarly, 30 days of data of HDD/CDD, occupancy, and day type value \((D)\) were used to get the tuned values of coefficients for the proposed equation-based systems. As for modified SVR and conventional SVR, 14 inputs (temperature, average dew points, relative humidity, specific humidity, indoor humidity, average wind speed, atmospheric pressure, average precipitation, insolation index and solar radiation, occupancy, normal weekdays/weekend/special holidays, HDD, CDD) were considered and 304 sets of data of 304 days were used for training and validations.

4.2. Effectiveness of Proposed Equation Based Prediction System over ANFIS, Random Forest, LSBoosting, and LSTM, Modified and Conventional SVR Methods

For all the prediction systems, as previously explained, randomly chosen 30 days of data were used for prediction and comparison purposes. For the ANFIS system, as previously explained, two inputs such as the temperature and P values were considered. For the equation-based systems, three inputs (HDD/CDD, occupancy, day type) and for other methods except modified and conventional SVR methods, three inputs (temperature, occupancy, day type) were considered and 304 sets of data of 304 days were used for training and validations. Since for all methods, occupancy and day type are common inputs, the data for the 30 predicted days were shown in Figure 8.

![Figure 8.](image.png)

**Figure 8.** (a) Occupant data and (b) day type data used for all the methods during prediction.

Figure 9 represents the comparison of prediction of energy consumptions by the proposed equations, ANFIS, random forest, LSBoosting, LSTM, modified and conventional SVR based prediction systems with actual energy consumption data. From the results, it is evident that the proposed equation-based prediction systems perform better as compared to all other systems.
Furthermore, the absolute percentage of error (\(\%\text{Err}\)), the absolute average error (\(\text{A.E}\)), root mean square error (\(\text{RMSE}\)), and mean average percentage error (\(\text{MAPE}\)) for the prediction systems have been calculated using (25), (26), (27), and (28), respectively.

The absolute percentage error shows the percentage of prediction error per day total consumption and helps determine the maximum error that occurs within the considered time period. The absolute average error predicts the average error of prediction from the actual consumption with the considered time periods. Similarly, the \(\text{RMSE}\) and \(\text{MAPE}\) shows the mean error and mean percentage of error over a considered time period. These error methods are very standard for the comparison of performance. The lower values of these errors mean the system predicts very close to the actual predictions. Therefore, these errors are used to evaluate the best system performance and these errors have been used as performance indices in this work.

\[
\%\text{Err} = \left| \frac{\text{Actual} - \text{Predicted}}{\text{Actual}} \right| \times 100, \tag{25}
\]

\[
\text{A.E} = \frac{1}{N} \sum_{i=1}^{N} |\text{Actual}_i - \text{Predicted}_i|, \tag{26}
\]

\[
\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\text{Actual}_i - \text{Predicted}_i)^2}, \tag{27}
\]

\[
\text{MAPE} = \frac{1}{N} \sum_{i=1}^{N} \%\text{Err}_i, \tag{28}
\]

where \(N = 30\) is used for Equations from (25) to (28). The percentage errors of proposed methods and other systems for predicting energy demands of chosen 30 days are shown in Figure 10.

Moreover, the average, root mean square and mean average percentage errors for all systems are shown in Table 8. From Table 8, it is evident that the average errors of equation-based prediction systems are less than those of ANFIS, random forest, LS boosting, LSTM, modified and conventional SVR based prediction systems. In this case, the proposed method shown in (1), (2), and (3) perform 29.75\%, 47.97\% and 48.63\% better, respectively, than the ANFIS system. The modified SVR performs 2.87\% better as compared to ANFIS system. However, the ANFIS system performs 106.8\%, 96.31\%, 109.01\%, and 71.31\% better as compared to random forest, LSBoosting, LSTM and conventional SVR methods, respectively.
Moreover, the RMSE values indicate that the equation-based systems proposed in (1) to (3) perform 48.72%, 50.83%, and 48.42% better, respectively, than the ANFIS system. The modified SVR performs 8.31% better as compared to the ANFIS system. However, the ANFIS system shows 44.18%, 59.38%, 54.87%, and 33.01% superior performance as compared to random forest, LSBoosting, LSTM, and conventional SVR methods, respectively. In addition, the equation-based systems perform 44.18%, 59.38%, 54.87%, and 33.01% superior performance as compared to random forest, LSBoosting, LSTM, modified and conventional SVR methods, respectively. Therefore, the proposed equation-based prediction systems perform better than other methods in all cases. Moreover, the errors of the ANFIS system are considered as the reference system for all performance improvement calculations mentioned above.

In addition to the RMSE error calculation, the sum of squares due to error (SSE), the coefficient of determination \( R^2 \) value is used to evaluate the goodness of fit statistics analysis [56]. The \( R^2 \) values are calculated based on the following Equation (29):

\[
R^2 = 1 - \frac{\text{SSE}}{\text{SST}} = 1 - \frac{\sum_{i=1}^{N} (y_i - \bar{y})^2}{\sum_{i=1}^{N} (y_i - \bar{y})^2},
\]  

Figure 10. Comparison of the percentage error in the prediction of all the methods for all days.

Table 8. Errors of predicted systems.

<table>
<thead>
<tr>
<th>System</th>
<th>AVG (kWh)</th>
<th>RMSE (kWh)</th>
<th>MAPE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest</td>
<td>5.09</td>
<td>6.07</td>
<td>35.60</td>
</tr>
<tr>
<td>LSBoost</td>
<td>4.79</td>
<td>6.71</td>
<td>22.19</td>
</tr>
<tr>
<td>LSTM</td>
<td>5.10</td>
<td>6.52</td>
<td>21.06</td>
</tr>
<tr>
<td>ANFIS</td>
<td>2.44</td>
<td>4.21</td>
<td>9.33</td>
</tr>
<tr>
<td>Modified SVR</td>
<td>2.37</td>
<td>3.86</td>
<td>12.14</td>
</tr>
<tr>
<td>Conventional SVR</td>
<td>4.18</td>
<td>5.60</td>
<td>28.04</td>
</tr>
<tr>
<td>Equation (1)</td>
<td>1.72</td>
<td>2.16</td>
<td>7.50</td>
</tr>
<tr>
<td>Equation (2)</td>
<td>1.27</td>
<td>2.07</td>
<td>6.05</td>
</tr>
<tr>
<td>Equation (3)</td>
<td>1.25</td>
<td>2.17</td>
<td>5.19</td>
</tr>
</tbody>
</table>

Moreover, the RMSE values indicate that the equation-based systems proposed in (1) to (3) perform 48.72%, 50.83%, and 48.42% better, respectively, than the ANFIS system. The modified SVR performs 8.31% better as compared to the ANFIS system. However, the ANFIS system shows 44.18%, 59.38%, 54.87%, and 33.01% superior performance as compared to random forest, LSBoosting, LSTM, and conventional SVR methods, respectively. In addition, the equation-based systems perform 19.62%, 35.21%, and 44.38% better, respectively, than the ANFIS system in terms of MAPE. Moreover, the ANFIS system performs 281.56%, 117.83%, 125.72%, 30.11%, and 170.42% better as compared to random forest, LSBoosting, LSTM, modified and conventional SVR methods, respectively. Therefore, the proposed equation-based prediction systems perform better than other methods in all cases. Moreover, the errors of the ANFIS system are considered as the reference system for all performance improvement calculations mentioned above.
where, $SST$ corresponds to sum of squares above the mean. Based on Equation (29), the $R^2$ value for the multiple linear regression optimization-based Equation (1) system is found to be 0.9804, which reflects that 98.04% of the total variation in the data ($N = 30$) are explained by the mentioned system. Moreover, $SSE$ and $SST$ values are found to be 139.867 and 7136.418, respectively.

5. Conclusions

This paper proposes new equations-based methods, based on HDD, CDD, occupancy and week/special days, for residential load forecasting. The performance of the proposed methods has been compared with that of the ANFIS, random forest, LSBoosting LSTM, modified and conventional SVR approaches. The forecasted energies by all methods are analyzed with actual energy consumption data for validation. The 304 days data are considered during training of the ANFIS, random forest, LSBoosting, LSTM, modified and conventional SVR systems. Moreover, 30 days of data of the same apartment are used for the prediction of all the methods. Based on the obtained simulation responses and performance indices, the following conclusions can be drawn.

1. The proposed equations-based methods are effective in predicting residential loads.
2. The proposed prediction systems require less computation and perform better than the ANFIS, random forest, LSBoosting, LSTM, and modified and conventional SVR systems. It is noteworthy that the energy consumption of a residential building depends upon the members living there with their habits, response to different environmental condition, mode of comfort, etc. Therefore, if the energy consumption is categorized based on HDD, CDD, number of occupants, the day type, the uncertainty of energy reduces much. From the Table 9 below, it is evident that if we consider the whole data range (bottom most row), the uncertainty of the system is high (12.91) in terms of standard deviation while the average energy consumption is 27.15 kWh. However, after dividing the data based on the conditions, it is evident that the average energy consumption is different than others and have much less uncertainty as compared to when considering the whole data. Moreover, this variation is seen because of various number of occupancies for a particular day and day type. That is why our proposed systems perform better than other considered systems. Moreover, our proposed systems do not require large data sets for training and sequential data for efficient prediction as it is required for LSTM but can efficiently predict any randomly chosen data.

Table 9. Average energy consumption and standard deviation under various conditions.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Average Energy Consumption (kWh)</th>
<th>Standard Deviation</th>
<th>%Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CDD \geq 17; O \geq 2; D \geq 1$</td>
<td>43.85</td>
<td>12.71</td>
<td>1.39</td>
</tr>
<tr>
<td>$CDD \geq 17; O \geq 2; D = 0$</td>
<td>33.59</td>
<td>6.80</td>
<td>47.32</td>
</tr>
<tr>
<td>$10 \leq CDD \leq 16; O \leq 1; D \leq 1$</td>
<td>12.31</td>
<td>4.39</td>
<td>65.99</td>
</tr>
<tr>
<td>$3 \leq CDD \leq 9; O \geq 3; D = 2$</td>
<td>38.20</td>
<td>6.15</td>
<td>50.81</td>
</tr>
<tr>
<td>$0 \leq CDD &lt; 3; O \leq 2; D \leq 2$</td>
<td>18.35</td>
<td>6.35</td>
<td>50.81</td>
</tr>
<tr>
<td>$HDD \geq 30; O \geq 0; D \geq 0$</td>
<td>38.53</td>
<td>8.06</td>
<td>37.57</td>
</tr>
<tr>
<td>$20 \leq HDD &lt; 30; O \geq 0; D \geq 0$</td>
<td>24.61</td>
<td>7.27</td>
<td>43.69</td>
</tr>
<tr>
<td>$10 \leq HDD &lt; 20; O \geq 0; D \geq 0$</td>
<td>21.24</td>
<td>6.75</td>
<td>47.71</td>
</tr>
<tr>
<td>$0.1 \leq HDD &lt; 10; O \geq 0; D \geq 0$</td>
<td>17.93</td>
<td>8.39</td>
<td>35.01</td>
</tr>
<tr>
<td>For all HDD, CDD, O, D</td>
<td>27.15</td>
<td>12.91</td>
<td>-</td>
</tr>
</tbody>
</table>

3. The proposed equation-based systems have the lowest absolute average errors (1.72%, 1.27%, 1.25%), root mean square errors (2.16, 2.07, 2.17) and mean average percentage errors (7.50%, 6.05%, 5.19%) compared to other methods.
4. The proposed equations-based systems can easily be implemented in real practice. In the near future, the performance of the equations-based prediction systems will be compared with other methods such as deep neural network, other new probabilistic prediction systems, etc. In addition, Bayesian optimization, which considers the data to have normal distribution, will be considered in the future work.
Author Contributions: Conceptualization, M.H.A. and S.M.M.A.; methodology, S.M.M.A.; software, S.M.M.A.; validation, S.M.M.A.; formal analysis, S.M.M.A.; investigation, S.M.M.A.; resources, S.M.M.A.; data curation, S.M.M.A.; writing—original draft preparation, S.M.M.A.; writing—review and editing, M.H.A.; visualization, S.M.M.A.; project administration, M.H.A.; funding acquisition, M.H.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References


Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).