**SHORT-TERM POWER LOAD FORECASTING FOR THE STAND-ALONE HYBRID POWER SYSTEM USING LSTM ARCHITECTURE**

Fazıl Efe Çavdara Erkan Dursunb

a Electric and Electronic Engineer, Marmara University, Bursa 16600, Turkey

b Electric and Electronic Department, Marmara University, Istanbul 34854, Turkey

# Abstract

Hybrid power systems are becoming more popular and needed more and more, but there are problems because these systems are nonlinear due to their dependence on weather conditions and are difficult to optimize. In this research, power generation forecasting of the stand-alone hybrid power system using Python was made with the LSTM architecture. The advantages of LSTM over other deep learning architectures in forecast these time series nonlinear systems are stated. In order to improve the optimization of the model, the window size, epoch size, batch size, learning rate and number of train and test data in the model were changed and their effects on the forecast, error rates and the running time of the model were observed. The lowest RMSE value in the model was observed as 0.0135 in model 5 in section 6.2. Effect of Epoch Size and the lowest Validation RMSE value in the model was observed as 0.0066 in model 5 in section 6.3. Effect of Batch Size and in model 1 in section 6.4 Effect of Learning Rate.

*Keywords: Sort-term power forecasting; Long Short-Term Memory Architecture; Stand-Alone Hybrid Power System*

# 1. Introduction

Renewable energy is a very important option for humanity trying to cope with the decrease of non-renewable energy resources day by day. Photovoltaic power generation (based on sunlight) is constitutes an important part of renewable energy power generation[1]. However, the variability and intermittent nature of hybrid power generation makes it difficult to integrate it into existing energy systems. Making an accurate and stable hybrid power generation forecast is a good way to solve this problem [2].

Hybrid power generation systems have received a lot of attention in the renewable energy sector in the past few decades and research on them has increased tremendously[3]; with the spread of PV and hybrid power generation day by day, significant environmental and economic benefits have been achieved, but due to the uncertain and non-linear characteristics of the hybrid power produced, it has brought new challenges as well as its benefits.[4]. These challenges include not only the hybrid power system being affected by natural conditions (meteorological), but also the high cost of installation as it has not yet become widespread enough.[5]. To overcome these challenges, estimating the power generation of hybrid systems like this is an efficient solution[6]. However, the dependence on metrology in hybrid systems causes instability and a nonlinear characteristic. Therefore, it becomes difficult to forecast the power generation of these systems[7].

Currently, a lot of research is being done on the power generation forecasting of hybrid systems, and new forecasting methods are emerging every day. We can examine these estimation methods in terms of time under three different headings; long-term power generation forecast, medium-term power generation forecast and short-term power generation forecast[8]. “Long-term forecasting looks ahead one month to one year. Medium-term forecasting considers a range of one week to one month, and short-term forecasting refers to a timeframe of one week or less.”[9].

Since 1990’s, computers have been used in all areas gradually and power generation prediction technology has emerged. With the development of machine learning and artificial neural network technologies, power generation forecasting has become more and more popular. The artificial neural network (ANN) model has very strong nonlinear modeling capabilities and is a data-driven nonlinear adaptive method[10]. Today, algorithms used for artificial neural networks are able to produce good predictions even in nonlinear functions, without knowing the relationship between the model to be predicted and the data [11]. Moreover, the support vector machine (SVM) originally employed by Vapnik and others of Bell Labs is extensively used in the domain of power load prediction and has been continuously improved by researchers.[12], [13].

The experimental results show that long short-term memory (LSTM ) and hybrid models work more accurately and significantly reduce error compared to other models, especially in time series power forecasting systems. The effectiveness of neural networks in power generation forecasting has been significantly validated. Artificial neural networks and deep learning (DL) technology have been used in the automation of many industries until today[14]. Compared to neural networks, there are hidden layers in deep learning and the word “deep” in deep learning corresponds to these layers. These layers can simulate neurons in the human brain, allowing the computer perceive problems like human. As a percentage, deep learning has become one of the most popular and attractive technologies for short-term electric power generation forecasting performance with convolution neural network (CNN), long short-term memory neural network (LSTM) and gated recurrent unit (GRU) algorithms [15],[16].

The final result of the deep learning network depends on the number of layers, input length, hyperparameters and the algorithm used, which affect the value and stability of the predictions. However, after a point, there is an inverse relationship between the number of layers and the values of other parameters and the training ability of the model. At first glance, we observe that while the number of layers is low, the accuracy of the prediction is also low and when we increase the number of layers, the accuracy increases. However, if we increase the number of layers when the forecast is at its best, we observe that the model deviates from the accuracy. That is why it is very important to correctly optimize the number of hidden layers and the other parameters[17].

In this paper proposes forecast a stand-alone hybrid power system with LSTM. This forecasting is help us how to optimize the hybrid system , make an energy saving and show the LSTM is why better than other deep learning methods for forecast a time series dataset. The flowchart of making a forecast with LSTM model shown in figure 1.1.

When we look at the literature, we do not see much power generation estimates of a stand-alone hybrid power system with a deep neural network architecture. In general, studies focused on PV or wind energy power generation forecasts are seen[7].

In these studies, the general purpose is to create new hybrid neural network architectures by using different deep neural network architectures on the same data set or by combining these architectures with each other and to observe their error rates and outputs [55], [56]. These hybrid neural network models also predict successfully. For example, this architecture can reduce the error rates between 65% and 35% in the estimation of the voltage magnitude of the power system with the CNN architecture, and the RNN-CNN hybrid architecture has been successful in reducing the RMSE error rate [57].

The different aspects of this study are that it uses the data of the stand-alone hybrid power system and instead of using more than one neural network architecture, a single architecture (LSTM) and the effects of the parameters in this deep neural network architecture on the error values are changed numerically and the differences between the actual value and the estimated value focused on showing it graphically.

Diagram

Description automatically generated

Fig. 1. 1. Flowchart for the build LSTM model and make forecasting for this research.

# 2. Deep Learning Methods

Deep learning is a subset of [machine learning](https://www.ibm.com/cloud/learn/machine-learning). It consists of at least two or more hidden layers and allows to learn large amounts of data. True-to-life predictions can also be made with a single hidden layer but adding enough hidden layers helps optimize and improve accuracy[18]. In this section we will see what Artificial Neural Network (ANN) is, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) and LSTM’s advantages. An architecture of deep neural network shown in figure 2.1.

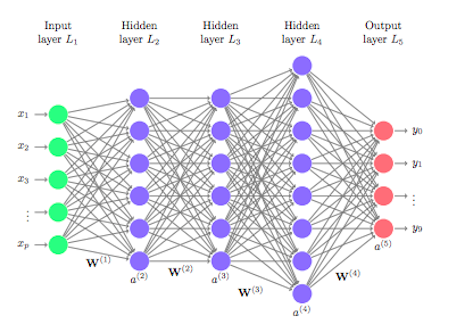


Fig. 2. 1. An example of deep neural network[19].

## 2.1. Artifical Neural Network (ANN)

An artificial neural network consists of processing units called neurons; model of an artificial neural network shown in figure 2.2. An artificial neuron tries to replicate the structure and behavior of the natural neuron. A neuron consists of inputs (dendrites), and one output (synapse via axon). The neuron has a function that determines the activation of the neuron[20].

A picture containing text, watch, device

Description automatically generated

Fig. 2. 2. Model of an artificial neuron [21].

The x values from x1 to xn are the inputs of the neuron, along with these inputs a bias value is added to the neuron. This bias value usually starts from 1. Values from W0 to Wn are weights. The weights are associated with the signal and multiplying a weight with the input gives us the strength of the signal. A neuron receives multiple inputs from other neurons and gives a single output.

The ANN architecture comprises of:

**a.** **input layer:** Receives the input values

**b.** **hidden layer(s):** A set of neurons between input and output layers. There can be single or multiple layers

**c. output layer:** Usually it has one neuron, and its output ranges between 0 and 1, that is, greater than 0 and less than 1. But multiple outputs can also be present.

The processing ability is stored in inter-unit connection strengths, called weights [3]. Input strength depends on the weight value. Weight value can be positive, negative or zero. Negative weight indicates that the signal is too weak or blocked. Zero weight indicates no connections between neurons. By adjusting the values of the weights we can get the outputs we need, this process is also called learning or training.[20].

Diagram, schematic

Description automatically generated

Fig. 2. 3. Neural network architecture [23].

## 2.2. Convolutional Neural Network (CNN)

Kunihiko Fukushima is the first person to introduce convolutional neural network(CNN), after that CNN was proposed by Yann LeCun. LeCun is combined CNN with back-propagation theory for the identify handwritten numbers and document identify. LeCun's system was used for the read handwritten documents and zip codes. CNN has convolutional and pooling layers. Convolutional layers are used to filter out useful information and have parameters that automatically optimize themselves to extract the most useful information. Pooling layers greatly reduces memory consumption and speeds up the model, thus enabling more convolutional layers to be used.[24].

Convolution, literally, is a function that defines the rule for how to compare two operations and produces the third operation. It is also expressed as the integral of the amount of agreement that occurs when the first two functions are over each other. Convolution is described by this formula:

(1)

A typical CNN architecture is described in Figure 2.4. Convolutional layers main task is doing mapping. This event means that pixels are completed to edges, motifs, parts, objects, and they in turn to scenes. The convolution layer uses activation functions to make the model more stable and successful. These can be hyperbolic tangent, sigmoid, or relu. However, as research show, the rectified linear unit (Relu) has achieved a more successful result compared to other activation functions. Definition of piecewise linear function:

(2)

Chart, diagram, waterfall chart

Description automatically generated

Fig. 2. 4. An example for typical CNN architecture[25].

## 2.3. Recurrent Neural Network (RNN)

A recurrent neural network (RNN) is a neural network that uses mostly sequential or time series data. This algorithm is used on problems such as temporal problems, language translation, speech recognition, captioning. Examples of usage areas are Siri and Google assistants. Like other neural networks, recurrent neural networks use training and test data. But unlike other neural networks, the most striking feature is that it has a memory. While the input and output are independent of each other in other neural networks used, RNN uses its memory to affect the current input and output. In other words, the output of the recurrent neural network depends on the previous inputs in it. For this reason, it is more intense to use recurrent neural networks instead of unidirectional neural networks in future prediction applications[26].

A picture containing text, clock

Description automatically generated

Fig. 2. 5. Recurrent Neural Network architecture[27].

Unlike feedforward networks, recurrent neural networks share the weight value in each layer, and these weights are continuously optimized during the backpropagation process and the gradient decline process to reinforce learning. This is one of the important features that distinguishes recurrent neural networks from others.

Recurrent neural networks make use of the time-backpropagation (BPTT) algorithm. BPTT trains itself by calculating the errors from the output layer to the input layer. This allows us to adjust the parameters in the model according to our wishes. While other neural networks do not need to collect errors since they do not share their parameters at every layer, one of the features that distinguishes BPTT from these neural networks is that it progresses by collecting errors and optimizing itself at every step.

Despite the above advantages, problems called bursting and disappearing gradients can be encountered in RNN algorithms. These problems are observed with the size of the gradient, which is the slope of the loss function. The problem with the disappearing gradient is that the gradient is too small because this continues until the weight parameters become unimportant, and when it reaches zero, the learning stops completely. In the exploding gradient problem, the opposite of the previous problem, the gradient value is very large and leads the model to instability. The complexity of the model can be reduced to avoid both of these situations. For example, reducing the number of hidden layers.[26].

### 

### 2.3.1. Common Activation Functions

**Sigmoid:**This is represented with the formula:

(3)

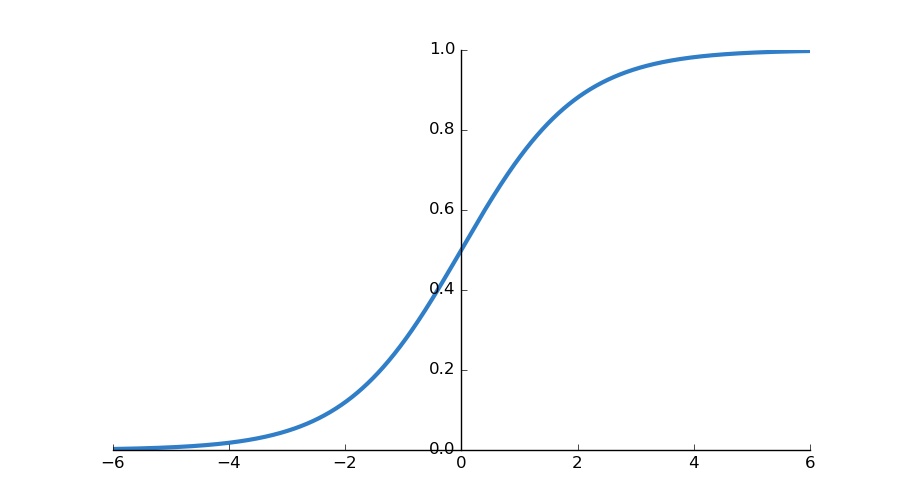


Figure 2. 6. Sigmoid Function’s graph[26].

**Tanh:**This is represented with the formula:

(4)

Chart

Description automatically generated

Figure 2. 7. Tanh function’s graph[26].

**Relu:**This is represented with the formula:

(5)

Chart, line chart

Description automatically generated

Fig. 2. 8. Relu function’s graph[26].

**2.3.2. Long Short-Term Memory (LSTM)**

The LSTM neural network is a model that is often used in time series and nonlinear signals processing[28]. “The LSTM is a special type of [RNN](https://www.sciencedirect.com/topics/engineering/recurrent-neural-network), first proposed by Hochreiter and Schmidhuber in 1997” [29], While LSTM has all the advantages of RNN, it is very popular because it also overcomes the vanishing gradient problem in RNN[30]. LSTM can easily learn long and short term data due to its special structure as shown in figure 2.9.[29]. The LSTM model has a memory cell with state at time t instead of neurons as in other neural networks. Apart from that, it controls the information flow in the memory with three gates, the input gate is , the output gate is and the forget gate is . While the forget gate resets the memory cell, the input and output gates provide the flow of information in the model. [31].

Diagram

Description automatically generated with medium confidence

Fig. 2. 9. LSTM structure [7].

Network layers in LSTM generate numbers between 0 and 1 to control the passing data. If the generated value is zero, the data pass is not allowed, while if the generated value is one, the data pass is completely allowed[32].

**Forgetting gate** : Produces a value between one and zero. A value of zero indicates that data will be completely separated and a value of zero indicates that data will be completely forgotten.

**Storage gate:** This layer consists of tanh and sigmoid layers. Its main purpose is to select the new data to be stored in the cell. While the sigmoid layer selects the values to be optimized, the tanh layer is responsible for generating the vector of the new values added.

**Input gate:** This layer is responsible for summarizing previous values to create new values [33].

The LSTM block can be described and the hidden state can be calculated by the following these equations[30].

(5)

(6)

(7)

(8)

(9)

“Where , and denote the forget gate, input gate and output gate, respectively; and are weight matrices; and are bias vectors; is the current input; is the output of LSTM at the previous time ; is the Sigmoid activation function; denotes the Hadamard production”[30].

Diagram, schematic

Description automatically generated

Fig. 2. 10. LSTM cell[34].

## 2.4. LSTM advantages

This section summarizes why LSTM is used in this paper [35];

* The LSTM can successfully capture patterns of both long-term (such as a few years) and short-term (a week, several hours) values.
* LSTM has the feature of categorizing different events that will affect the output.
* The LSTM model can receive data of different lengths as input
* LSTM can be used in many different industries.
* Some factors have non-linear effects with the desired output. LSTM can learn these non-linear effects thanks to the gates inside.
* LSTM contains more parameters than RNN, which allows more calculations.

Because of those, LSTM is used for the time series forecasting in this paper.

# 3. Python

Python is an object-oriented, interpretative, unitary and interactive high-level programming language developed by Guido Van Rossum in 1991. One of its most important features is its readability and ease of writing. In this way, the maintenance cost of the program is also low. It has a very extensive library, is open to development and is free.[36].

## 3.1. Advantages of Python

It is very easy for even a beginner to write a "hello world" program compared to other programming languages. At the end of the line, it doesn't need special characters such as “;”. The syntax of Python code is very comfortable and flexible, it can work with other operating systems, it has a lot of documentation, and the job opportunity is very wide, it can be used in almost every industry. E.g., data analysis and visualization, machine learning, game development, web applications, user interface development... As a result, python is one of the most preferred coding languages due to its readability and ease of implementation. In 2019, it was the most used language according to Statista's data[37].

## 3.2. Keras

Keras is an application programming interface (API) specialized for deep learning, written in Python, running on TensorFlow, a machine learning platform[38].

**Simple --** Keras reduces the developer's cognitive load to increase the developer's focus on key parts of the program[38].

**Flexible --** Keras shows complexity incrementally so the user progresses by understanding and building on simple workflows[38].

**Powerful --** It is also used by very large organizations and companies in the world. For example YouTube, NASA… [38].

# 4. Stand-Alone Hybrid Power System

A stand-alone hybrid power system (SAHPS) is an energy system that includes at least two types of generators. These generators can be photovoltaic (PV), wind energy conversion system (WECS) or diesel generator.[39]. Today, rechargeable, battery-powered or non-battery types and hybrid combinations of these generators are highly developed, and their use is being researched and supported to make the cost even more affordable[40]. In order to obtain electricity safely from these systems, optimization must be done very well, in a well-optimized hybrid system, besides safe electricity, the cost is also appropriate compared to other systems[41]. However, the design problem of hybrid power systems is difficult and cannot be solved by trial-and-error method. It has many nonlinear variables and the load demands are high. That's why optimization is important.[42],[43].

Diagram, schematic

Description automatically generated

Fig. 4. 1. Configuration of standalone hybrid power system.

One of the most important features of the standalone hybrid power system is that it can operate completely independently in an optimal environment.[44]. In addition, a well-designed and optimized stand-alone hybrid power system is expected to work stably without causing problems with other devices. In short, it should have plug and play feature[45].

It may be necessary to use different energy sources and different ways of integrating to create a stand-alone hybrid power system, these can be dc-coupled, ac-coupled and hybrid. [46].

## 4.1. DC-Coupled Systems

The coupling scheme of the DC connected system is simple and does not need to be synchronized with an AC system, the block diagram of the system is shown in figure 4.2. The system is connected to the DC bus with suitable converters and the system is flexible so it can be connected to another DC, AC or hybrid system with suitable converters. However, when connected to an AC system, it must be at a frequency of 50 or 60 Hz. However, in addition to these advantages, the DC-connected system also has some disadvantages. For example, if the converters in the system are damaged or not active, the system cannot produce AC power, which can be solved with parallel connected inverters.[47].

Diagram

Description automatically generated

Fig. 4. 2. Schematic diagram DC-coupled hybrid energy system[46].

## 4.2. AC-Coupled System

We can examine AC connected power systems in two categories, which are shown in figure 4.3 as (a) and (b). Figure (a) is a power frequency coupled AC system (PFAC) and it consists of integrating different energy sources into the AC bus, which requires a coupling inductor between the AC and the bus. The system shown in (b) in Figure 4.3 is connected to high frequency AC (HFAC), where the loads are connected between the HFAC and its bus. The scope of use is quite wide, such as submarines, ships and space stations.

In both of these systems, AC and DC outputs can be obtained with suitable converters as in the DC connected system.[46].

Diagram

Description automatically generated

Fig. 4. 3. Schematic diagram AC-coupled hybrid energy system. (a) is power frequency AC system(PFAC), (b) is high frequency AC system (HFAC)[46].

## 4.3. Hybrid-Coupled System

In a hybrid coupled system (shown in figure 4.4), various sources can be connected to both DC and AC buses, and this makes the system more stable, highly efficient and at lower costs. However, as a disadvantage to these situations, the complexity of the system increases and it becomes more difficult both in terms of design and optimization.[46].

Diagram

Description automatically generated

Fig. 4. 4. Schematic diagram of hybrid-coupled (DC bus and power frequency AC bus) hybrid energy system[46].

# 5. LSTM Model

The input layer of each LSTM model must be three-dimensional, like (features, time steps, samples).

The three dimensions of this input are:

* Samples : One sequence is one sample. A batch is comprised of one or more samples.
* Time Steps : One time step is one point of observation in the sample.
* Features : One feature is one observation at a time step.

Even if the model to be used will predict a single value or a single set of features while estimating, the input of LSTM must be three-dimensional and the parameters in this "input\_shape" can be changed manually and the model can be used in any direction.[48]. The input and output shape of the LSTM model in this research is shown in figure 5.1. The LSTM model and the number of layers to be used in the research are shown in figure 5.2.

Text, letter

Description automatically generated

Fig. 5. 1. Input and output shape in LSTM model.

Table

Description automatically generated

Fig. 5. 2. LSTM model for this research.

## 5.1. ADAM Optimizer

The adaptive moment estimation (ADAM) optimizer is a very popular and successful optimizer used in deep learning. In short, it calculates adaptive learning rates for each parameter in it and prefers flat minimums on the error surface. Unlike other optimizers, keeps exponentially decreasing average of historical gradients, stores gradients like Adadelta and RMSprop exponentially.[49]. The descending averages of the and gradients are calculated as:

(10)

(11)

is the mean of the first moment and is the uncentered variance estimate of the second moment. The and in the equations tend to go towards one because the decay rates in the first timesteps are quite small.[50].

Eliminate biases using first and second moment estimators:

(12)

(13)

We then use this equation to update the parameters in Adadelta and RMSprop:

(14)

The creators of the ADAM optimizer suggest a value of 0.99 for , 0.999 for , and for the value, experiments show that the optimizer works well with other learning algorithms at these values. However, the user can shape these values according to his own wishes[50].

## 5.2. Mean Squared Error

The mean square error (MSE) evaluates the mean square difference between observed and predicted values. In other words, it measures the amount of error of statistical models and predictions, if there is no error, its value is zero, and the value of MSE increases as the error increases.[51].

Chart, scatter chart

Description automatically generated

Fig. 5. 3. Regression line with predictions[52].

As can be seen in Figure 5.3, the closer the data points are to the regression line, the less errors there will be, the closer the model's predictions will be to the truth, and the lower the value of the MSE.

Formula for mean square error (MSE):

(15)

Where:

is the observed value.

is the corresponding predicted value.

is the number of observations.

In words, finding the MSE is divided by the number of observations by squaring the difference between the true value and the estimated value for all values similar to the variance.[51].

## 5.3. Root Mean Squared Error

Root mean square error (RMSE) is a widely used and very successful general prediction error measure in deep learning. The total error is found by averaging the square of the error.[53].

(16)

The , and used in the equation are the observations, the predicted value of one variable and the number of observations, respectively. One disadvantage of RMSE is that it only compares the errors of the model for one variable, not between variables. This is because it depends on the scale.[53].

# 6. Results

In this section we change the parameters of LSTM model (window size, epochs, batch size, learning rate and number of train and test data) and we observe their effect on the error rates of the model and the graphs between the prediction and the actual results and running time. The LSTM layer numbers that we will use while doing these experiments are shown in figure 6.1. Later in the experiment, the number of trained parameters will change as window size and number of train and test data change.

Table

Description automatically generated

Fig. 6. 1. Used LSTM model layers in this section.

## 6.1 Effects of Window Size

After training the proposed models for to observe window size effect, it is used to test the performance of the forecasting model to predict AC output of the hybrid power system. The effect of window size on error rates and running time is shown in table 1.

* Number of train data = 13,000
* Number of test data = 2,189
* Optimizer = ADAM
* Learning rate = 0.0001
* Batch size = 2
* Epochs = 10

Table 1. Changes in runtime and error values when we change window size.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Window  size | Running speed | MSE | RMSE | Validation loss | Validation RMSE |
| 1 | 3 | 256s 38ms | 2.8894e-04  (0.00028894) | 0.0170 | 49704e-05  (0.000049704) | 0.0071 |
| 2 | 5 | 337s 52ms | 2.5837e-04 | 0.0161 | 4.8977e-05 | 0.0070 |
| 3 | 10 | 549s 83ms | 2.5217e-04 | 0.0159 | 4.4341e-05 | 0.0067 |
| 4 | 20 | 862s 132ms | 2.6392e-04 | 0.0162 | 4.9118e-05 | 0.0070 |
| 5 | 50 | 1891s 292ms | 2.5232e-04 | 0.0159 | 5.9389e-05 | 0.0077 |

**Model 1:** When window size is 3, without changing other parameters the actual and predicted values graph shown in figure 6.2. The blue line represents actual values and orange line represent predicted values.

Shape

Description automatically generated

Fig. 6.2. Plot of Ac output power actual and predicted values when window size is 3.

**Model 2:** When window size is 5, without changing other parameters the actual and predicted values graph shown in figure 6.3. The blue line represents actual values and orange line represent predicted values.

Diagram

Description automatically generated with low confidence

Fig. 6. 3. Plot of Ac output power actual and predicted values when window size is 5.

**Model 3:**  When window size is 10, without changing other parameters the actual and predicted values graph shown in figure 6.4. The blue line represents actual values and orange line represent predicted values.

Chart

Description automatically generated with low confidence

Fig. 6. 4. Plot of Ac output power actual and predicted values when window size is 10.

**Model 4 :** When window size is 20, without changing other parameters the actual and predicted values graph shown in figure 6.5. The blue line represents actual values and orange line represent predicted values.

Diagram

Description automatically generated with low confidence

Fig. 6. 5. Plot of Ac output power actual and predicted values when window size is 20.

**Model 5:** When window size is 50, without changing other parameters the actual and predicted values graph shown in figure 6.6. The blue line represents actual values and orange line represent predicted values.

Chart, histogram

Description automatically generated

Fig. 6. 6. Plot of Ac output power actual and predicted values when window size is 50.

## 6.2 Effect of Epoch Size

After training the proposed models, for to observe effect of epoch size, it is used to test the performance of the forecasting model to predict AC output of the hybrid power system. The effect of epoch size on error rates and running time is shown in table2.

* Number of train data = 13,000
* Number of test data = 2,189
* Optimizer = ADAM
* Learning rate = 0.0001
* Batch size = 2
* Window size = 5

Table 2. Changes in runtime and error values when we change epoch size.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Epochs | Running speed | MSE | RMSE | Validation loss | Validation RMSE |
| 1 | 3 | 142s 23ms | 4.3715e-04 | 0.0209 | 4.8318e-05 | 0.0070 |
| 2 | 10 | 354s 70ms | 2.7351e-04 | 0.0165 | 4.0382e-05 | 0.0071 |
| 3 | 50 | 986s 250ms | 2.2631e-04 | 0.0150 | 4.9356e-05 | 0.0070 |
| 4 | 120 | 3770s 345ms | 2.0338e-04 | 0.0143 | 4.9075e-05 | 0.0070 |
| 5 | 200 | 6201s 45ms | 1.8333e-04 | 0.0135 | 6.4139e-05 | 0.0080 |

**Model 1:** When epoch size is 3, without changing other parameters the actual and predicted values graph shown in figure 6.7. The blue line represents actual values and orange line represent predicted values.

Chart, line chart

Description automatically generated

Fig. 6. 7. Plot of Ac output power actual and predicted values when epoch size is 3.

**Model 2:** When epoch size is 10, without changing other parameters the actual and predicted values graph shown in figure 6.8. The blue line represents actual values and orange line represent predicted values.

A picture containing line chart

Description automatically generated

Fig. 6. 8. Plot of Ac output power actual and predicted values when epoch size is 10.

**Model 3:** When epoch size is 50, without changing other parameters the actual and predicted values graph shown in figure 6.9. The blue line represents actual values and orange line represent predicted values.

A picture containing chart

Description automatically generated

Fig. 6. 9. Plot of Ac output power actual and predicted values when epoch size is 50.

**Model 4:** When epoch size is 120, without changing other parameters the actual and predicted values graph shown in figure 6.10. The blue line represents actual values and orange line represent predicted values.

Chart

Description automatically generated with medium confidence

Fig. 6. 10. Plot of Ac output power actual and predicted values when epoch size is 120.

**Model 5:** When epoch size is 200, without changing other parameters the actual and predicted values graph shown in figure 6.11. The blue line represents actual values and orange line represent predicted values.

Chart

Description automatically generated

Fig. 6. 11. Plot of Ac output power actual and predicted values when epoch size is 200.

## 6.3 Effect of Batch Size

After training the proposed models, for to observe effect of batch size, it is used to test the performance of the forecasting model to predict AC output of the hybrid power system. The effect of batch size on error rates and running time is shown in table3.

“Batch size is the number of samples that are passed to the network at once[54].”

* Number of train data = 13,000
* Number of test data = 2,189
* Optimizer = ADAM
* Learning rate = 0.0001
* Epochs = 10
* Window size = 5

Table 3. Changes in runtime and error values when we change batch size.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Batch size | Running Speed | MSE | RMSE | Validation Loss | Validation RMSE |
| 1 | 1 | 594s 40ms | 2.5659e-04 | 0.0160 | 4.5500e-05 | 0.0067 |
| 2 | 3 | 268s 70ms | 2.3234e-04 | 0.0152 | 4.5777e-05 | 0.0068 |
| 3 | 5 | 225s 88ms | 2.2698e-04 | 0.0151 | 4.5378e-05 | 0.0067 |
| 4 | 10 | 135s 88ms | 2.2232e-04 | 0.0149 | 4.4408e-05 | 0.0067 |
| 5 | 20 | 77s 121ms | 2.1751e-04 | 0.0147 | 4.4133e-05 | 0.0066 |

**Model 1:** When batch size is 1, without changing other parameters the actual and predicted values graph shown in figure 6.12. The blue line represents actual values and orange line represent predicted values.

Chart, line chart

Description automatically generated

Fig. 6. 12. Plot of Ac output power actual and predicted values when batch size is 1.

**Model 2:** When batch size is 3, without changing other parameters the actual and predicted values graph shown in figure 6.13. The blue line represents actual values and orange line represent predicted values.

Chart

Description automatically generated

Fig. 6. 13. Plot of Ac output power actual and predicted values when batch size is 3.

**Model 3**: When batch size is 5, without changing other parameters the actual and predicted values graph shown in figure 6.14. The blue line represents actual values and orange line represent predicted values.

Chart, line chart

Description automatically generated

Fig. 6. 14. Plot of Ac output power actual and predicted values when batch size is 5.

**Model 4:** When batch size is 10, without changing other parameters the actual and predicted values graph shown in figure 6.15. The blue line represents actual values and orange line represent predicted values.

Chart, line chart

Description automatically generated

Fig. 6. 15. Plot of Ac output power actual and predicted values when batch size is 10.

**Model 5:** When batch size is 20, without changing other parameters the actual and predicted values graph shown in figure 6.16. The blue line represents actual values and orange line represent predicted values.

Chart, line chart

Description automatically generated

Fig. 6. 16. Plot of Ac output power actual and predicted values when batch size is 20.

## 6.4 Effect of Learning Rate

After training the proposed models, for to observe effect of learning rate, it is used to test the performance of the forecasting model to predict AC output of the hybrid power system. The effect of learning rate on error rates and running time is shown in table4.

* Number of train data = 13,000
* Number of test data = 2,189
* Optimizer = ADAM
* Batch size = 2
* Epochs = 10
* Window size = 5

Table 4. Changes in runtime and error values when we change learning rate.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Learning Rate | Running Speed | MSE | RMSE | Validation Loss | Validation RMSE |
| 1 | 0.00001 | 359s 59ms | 2.0809e-04 | 0.0144 | 4.3456e-05 | 0.0066 |
| 2 | 0.0001 | 330s 54ms | 2.2451e-04 | 0.0150 | 4.7831e-05 | 0.0069 |
| 3 | 0.001 | 402s 55ms | 2.4599e-04 | 0.0157 | 6.4561e-05 | 0.0080 |
| 4 | 0.01 | 336s 50ms | 3.6914e-04 | 0.0192 | 8.8467e-05 | 0.0094 |
| 5 | 0.1 | 395s 64ms | 0.0239 | 0.1547 | 0.0162 | 0.1273 |

**Model 1:** When learning rate is 0.00001, without changing other parameters the actual and predicted values graph shown in figure 6.17. The blue line represents actual values and orange line represent predicted values.

Chart, line chart

Description automatically generated

Fig. 6. 17. Plot of Ac output power actual and predicted values when learning rate is 0.00001.

**Model 2:** When learning rate is 0.0001, without changing other parameters the actual and predicted values graph shown in figure 6.18. The blue line represents actual values and orange line represent predicted values.

Chart

Description automatically generated

Fig. 6. 18. Plot of Ac output power actual and predicted values when learning rate is 0.0001.

**Model 3:** When learning rate is 0.001, without changing other parameters the actual and predicted values graph shown in figure 6.19. The blue line represents actual values and orange line represent predicted values.

Shape

Description automatically generated with medium confidence

Fig. 6.19. Plot of Ac output power actual and predicted values when learning rate is 0.001.

**Model 4:** When learning rate is 0.01, without changing other parameters the actual and predicted values graph shown in figure 6.20. The blue line represents actual values and orange line represent predicted values.

Shape

Description automatically generated with medium confidence

Fig. 6. 20. Plot of Ac output power actual and predicted values when learning rate is 0.01.

**Model 5:** When learning rate is 0.1, without changing other parameters the actual and predicted values graph shown in figure 6.21. The blue line represents actual values and orange line represent predicted values.

Chart

Description automatically generated

Fig. 6.21. Plot of Ac output power actual and predicted values when learning rate is 0.1.

## 6.5 Effect of Number of Train and Test Data

After training the proposed models, for to observe effect of number of train and test data, it is used to test the performance of the forecasting model to predict AC output of the hybrid power system. The effect of number of train and test data on error rates and running time is shown in table5.

* Optimizer = ADAM
* Learning rate = 0.0001
* Batch size = 2
* Epochs = 10
* Window size = 5

Table 5. Changes in runtime and error values when we change number of train and test data.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Number of Train | Number of Test | Running Time | MSE | RMSE | Validation Loss | Validation RMSE |
| 1 | 14000 | 1189 | 368s 52ms | 2.4187e-04 | 0.0156 | 5.0523e-05 | 0.0071 |
| 2 | 12000 | 3189 | 331s 55ms | 4.0315e-04 | 0.0184 | 8.4859e-05 | 0.0092 |
| 3 | 10000 | 5189 | 371s 77ms | 3.3877e-04 | 0.0201 | 2.6965e-04 | 0.0164 |
| 4 | 7000 | 8189 | 173s 48ms | 5.1344e-04 | 0.0227 | 2.8048e-06 | 0.00170 |
| 5 | 5000 | 10189 | 273s 109ms | 6.7441e-04 | 0.0260 | 3.9486e-04 | 0.0199 |

**Model 1:** When number of train data is 14000 and test data is 1189, without changing other parameters the actual and predicted values graph shown in figure 6.22. The blue line represents actual values and orange line represent predicted values.

Shape

Description automatically generated with medium confidence

Fig. 6. 22. Plot of Ac output power actual and predicted values when number of train data is 14000 and test data is 1189.

**Model 2:** When number of train data is 12000 and test data is 3189, without changing other parameters the actual and predicted values graph shown in figure 6.23. The blue line represents actual values and orange line represent predicted values.

A picture containing shape

Description automatically generated

Fig. 6. 23. Plot of Ac output power actual and predicted values when number of train data is 12000 and test data is 3189.

**Model 3:** When number of train data is 10000 and test data is 5189, without changing other parameters the actual and predicted values graph shown in figure 6.24. The blue line represents actual values and orange line represent predicted values.

A picture containing shape

Description automatically generated

Fig. 6. 24. Plot of Ac output power actual and predicted values when number of train data is 10000 and test data is 5189.

**Model 4:** When number of train data is 7000 and test data is 8189, without changing other parameters the actual and predicted values graph shown in figure 6.25. The blue line represents actual values and orange line represent predicted values.

A picture containing shape

Description automatically generated

Fig. 6. 25. Plot of Ac output power actual and predicted values when number of train data is 7000 and test data is 8189.

**Model 5:** When number of train data is 5000 and test data is 10189, without changing other parameters the actual and predicted values graph shown in figure 6.26. The blue line represents actual values and orange line represent predicted values.

Shape

Description automatically generated

Fig. 6. 26. Plot of Ac output power actual and predicted values when number of train data is 5000 and test data is 10189.

# 7. Conclusions

In this research, with the literature review and tests, the power generation forecasting of the short-term data taken from an independent hybrid power system was made using the LSTM architecture. By changing the parameters in the model, it has been observed which parameter affects the model and how.

Hybrid power systems are multivariate and non-linear. Therefore, estimating the power generation of these systems and optimizing the system well are important in terms of obtaining both economic and realistic results. Therefore, in this research, besides the power generation forecasting of the standalone hybrid power system, the effects of the parameters in this model on the output of the model were observed both with MSE, RMSE, Validation Loss and Validation RMSE error values, and the actual value and the estimated value of the model were shown graphically. Table 6. shows how the running time and error values change when we increase the values of the parameters.

Table 6. changes in running time and error parameters.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameters with increasing value | Running time | MSE | RMSE |
| Window size | Increase | Decrease until 10 | Decrease until 10 |
| Epoch size | Increase | Decrease | Decrease |
| Batch size | Decrease | Decrease | Decrease |
| Learning rate | - | Increase | Increase |
| Number of test/train data | - | Increase | Increase |

As can be seen in Table 6. and section 6, there are no exact parameters for a model with the fastest and least error values. There is a trade-off between them, for example, as epoch size increase, the error values decrease, but the model's running speed decreases. As seen in sections 6.4 and 6.5, increasing learning rate and number of test/train values increases the error rates between the true value and the predicted value and does not directly affect the running speed of the model. As can be seen in section 6.1, with the increase of window size, the error values decrease for a while, but then it starts to increase, it is observed that the model also decreases the running speed. Increasing batch size, as seen in section 6.3, speeds up the system, but increases the difference between the actual value and the estimated value.

As a result, the power generation forecasting of the stand-alone hybrid power system was made with LSTM. For the optimization of the model, the parameters were changed and presented as both tables and figures. However, when forecast a hybrid power generation, there is no one-size-fits-all solution. For different systems and models, the parameters should be changed and the most suitable ones for that model should be selected.

# References

[1] “Progress of solar photovoltaic in ASEAN countries: A review - ScienceDirect”. https://www.sciencedirect.com/science/article/pii/S1364032115002804 (erişim Oca. 05, 2022).

[2] R. Blaga, A. Sabadus, N. Stefu, C. Dughir, M. Paulescu, ve V. Badescu, “A current perspective on the accuracy of incoming solar energy forecasting”, *Prog. Energy Combust. Sci.*, c. 70, ss. 119-144, Oca. 2019, doi: 10.1016/j.pecs.2018.10.003.

[3] F. Creutzig, P. Agoston, J. C. Goldschmidt, G. Luderer, G. Nemet, ve R. C. Pietzcker, “The underestimated potential of solar energy to mitigate climate change”, *Nat. Energy*, c. 2, sy 9, ss. 1-9, Ağu. 2017, doi: 10.1038/nenergy.2017.140.

[4] G. Stein ve T. M. Letcher, “15 - Integration of PV Generated Electricity into National Grids”, içinde *A Comprehensive Guide to Solar Energy Systems*, T. M. Letcher ve V. M. Fthenakis, Ed. Academic Press, 2018, ss. 321-332. doi: 10.1016/B978-0-12-811479-7.00015-4.

[5] “Short-term photovoltaic power forecasting using Artificial Neural Networks and an Analog Ensemble - ScienceDirect”. https://www.sciencedirect.com/science/article/pii/S0960148117301386 (erişim Oca. 05, 2022).

[6] X. G. Agoua, R. Girard, ve G. Kariniotakis, “Short-Term Spatio-Temporal Forecasting of Photovoltaic Power Production”, *IEEE Trans. Sustain. Energy*, c. 9, sy 2, ss. 538-546, Nis. 2018, doi: 10.1109/TSTE.2017.2747765.

[7] P. Li, K. Zhou, X. Lu, ve S. Yang, “A hybrid deep learning model for short-term PV power forecasting”, *Appl. Energy*, c. 259, s. 114216, ubat 2020, doi: 10.1016/j.apenergy.2019.114216.

[8] “Machine learning methods for solar radiation forecasting: A review - ScienceDirect”. https://www.sciencedirect.com/science/article/pii/S0960148116311648 (erişim Oca. 05, 2022).

[9] M. Pierro *vd.*, “Deterministic and Stochastic Approaches for Day-Ahead Solar Power Forecasting”, *J. Sol. Energy Eng.*, c. 139, sy 2, Kas. 2016, doi: 10.1115/1.4034823.

[10] X. Zhang, J. Wang, ve Y. Gao, “A hybrid short-term electricity price forecasting framework: Cuckoo search-based feature selection with singular spectrum analysis and SVM”, *Energy Econ.*, c. 81, ss. 899-913, Haz. 2019, doi: 10.1016/j.eneco.2019.05.026.

[11] J. Liu, X. Liu, ve B. T. Le, “Rolling Force Prediction of Hot Rolling Based on GA-MELM”, *Complexity*, c. 2019, s. e3476521, Haz. 2019, doi: 10.1155/2019/3476521.

[12] G.-F. Fan, Y.-H. Guo, J.-M. Zheng, ve W.-C. Hong, “A generalized regression model based on hybrid empirical mode decomposition and support vector regression with back-propagation neural network for mid-short-term load forecasting”, *J. Forecast.*, c. 39, sy 5, ss. 737-756, 2020, doi: 10.1002/for.2655.

[13] “Research on Ultra-Short-Term Load Forecasting of Distribution Network Based on Fuzzy Clustering and RBF Neural Network - ProQuest”. https://www.proquest.com/docview/2465722557?pq-origsite=gscholar&fromopenview=true (erişim Oca. 05, 2022).

[14] A. Almalaq ve G. Edwards, “A Review of Deep Learning Methods Applied on Load Forecasting”, içinde *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Ara. 2017, ss. 511-516. doi: 10.1109/ICMLA.2017.0-110.

[15] Y. Jin, H. Guo, J. Wang, ve A. Song, “A Hybrid System Based on LSTM for Short-Term Power Load Forecasting”, *Energies*, c. 13, sy 23, Art. sy 23, Oca. 2020, doi: 10.3390/en13236241.

[16] X. He, Y. Nie, H. Guo, ve J. Wang, “Research on a Novel Combination System on the Basis of Deep Learning and Swarm Intelligence Optimization Algorithm for Wind Speed Forecasting”, *IEEE Access*, c. 8, ss. 51482-51499, 2020, doi: 10.1109/ACCESS.2020.2980562.

[17] Z. He, Y. Chen, Z. Shang, C. Li, L. Li, ve M. Xu, “A novel wind speed forecasting model based on moving window and multi-objective particle swarm optimization algorithm”, *Appl. Math. Model.*, c. 76, ss. 717-740, Ara. 2019, doi: 10.1016/j.apm.2019.07.001.

[18] “What is Deep Learning?”, 16 Ağustos 2021. https://www.ibm.com/cloud/learn/deep-learning (erişim Oca. 05, 2022).

[19] “It’s Deep Learning Times: A New Frontier of Data | by Sunpark | Towards Data Science”. https://towardsdatascience.com/its-deep-learning-times-a-new-frontier-of-data-a1e9ef9fe9a8 (erişim Oca. 05, 2022).

[20] K. Shiruru, “AN INTRODUCTION TO ARTIFICIAL NEURAL NETWORK”, *Int. J. Adv. Res. Innov. Ideas Educ.*, c. 1, ss. 27-30, Eyl. 2016.

[21] “What is a perceptron in neural networks?”, *Quora*. https://www.quora.com/What-is-a-perceptron-in-neural-networks (erişim Oca. 05, 2022).

[22] M. Toprak, “Activation Functions for Deep Learning”, *Medium*, 14 Haziran 2020. https://medium.com/@toprak.mhmt/activation-functions-for-deep-learning-13d8b9b20e (erişim Oca. 05, 2022).

[23] “Neural Network Architectures”, *Manning*, 22 Temmuz 2019. https://freecontent.manning.com/neural-network-architectures/ (erişim Oca. 05, 2022).

[24] Department of Electrical and Electronics Engineering, Firat University, Elazığ, Turkey *vd.*, “AN OVERVIEW OF POPULAR DEEP LEARNING METHODS”, *Eur. J. Tech.*, c. 7, sy 2, ss. 165-176, Ara. 2017, doi: 10.23884/ejt.2017.7.2.11.

[25] M. Aamir, Z. Rahman, W. Abro, M. Tahir, ve S. Mustajar, “An Optimized Architecture of Image Classification Using Convolutional Neural Network”, *Int. J. Image Graph. Signal Process.*, c. 11, ss. 30-39, Eki. 2019, doi: 10.5815/ijigsp.2019.10.05.

[26] “What are Recurrent Neural Networks?”, 07 Nisan 2021. https://www.ibm.com/cloud/learn/recurrent-neural-networks (erişim Oca. 05, 2022).

[27] Zhu, Juncheng, Zhile Yang, Monjur Mourshed, Yuanjun Guo, Yimin Zhou, Yan Chang, Yanjie Wei, ve Shengzhong Feng. “Electric Vehicle Charging Load Forecasting: A Comparative Study of Deep Learning Approaches”. *Energies* 12 (13 Temmuz 2019): 2692. <https://doi.org/10.3390/en12142692>.

[28] “Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM - ScienceDirect”. https://www.sciencedirect.com/science/article/pii/S0360544218302056 (erişim Oca. 05, 2022).

[29] S. Hochreiter ve J. Schmidhuber, “Long Short-Term Memory”, *Neural Comput.*, c. 9, sy 8, ss. 1735-1780, Kas. 1997, doi: 10.1162/neco.1997.9.8.1735.

[30] “A process optimization strategy of a pulsed-spray fluidized bed granulation process based on predictive three-stage population balance model - ScienceDirect”. https://www.sciencedirect.com/science/article/pii/S0032591017310288 (erişim Oca. 05, 2022).

[31] F. A. Gers, J. Schmidhuber, ve F. Cummins, “Learning to Forget: Continual Prediction with LSTM”, *Neural Comput.*, c. 12, sy 10, ss. 2451-2471, Eki. 2000, doi: 10.1162/089976600300015015.

[32] S. Siami-Namini ve A. S. Namin, “Forecasting Economics and Financial Time Series: ARIMA vs. LSTM”, *ArXiv180306386 Cs Q-Fin Stat*, Mar. 2018, Erişim: 07 Ocak 2022. [Çevrimiçi]. Erişim adresi: http://arxiv.org/abs/1803.06386

[33] Y. Wang, S. Zhu, ve C. Li, “Research on Multistep Time Series Prediction Based on LSTM”, içinde *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, Eki. 2019, ss. 1155-1159. doi: 10.1109/EITCE47263.2019.9095044.

[34] “Lake Level Prediction using Feed Forward and Recurrent Neural Networks”. Erişim 11 Mayıs 2022. https://www.researchgate.net/publication/332766508\_Lake\_Level\_Prediction\_using\_Feed\_Forward\_and\_Recurrent\_Neural\_Networks/figures.

[35] “Understanding LSTM in Time Series Forecasting - PredictHQ”. https://www.predicthq.com/events/lstm-time-series-forecasting (erişim Oca. 08, 2022).

[36] “What is Python? Executive Summary”, *Python.org*. https://www.python.org/doc/essays/blurb/ (erişim Oca. 08, 2022).

[37] A. Bogdanchikov, M. Zhaparov, ve R. Suliyev, “Python to learn programming”, *J. Phys. Conf. Ser.*, c. 423, s. 012027, Nis. 2013, doi: 10.1088/1742-6596/423/1/012027.

[38] K. Team, “Keras documentation: About Keras”. https://keras.io/about/ (erişim Oca. 09, 2022).

[39] M. K. Deshmukh ve S. S. Deshmukh, “Modeling of hybrid renewable energy systems”, *Renew. Sustain. Energy Rev.*, c. 12, sy 1, ss. 235-249, Oca. 2008, doi: 10.1016/j.rser.2006.07.011.

[40] C. Dennis Barley ve C. Byron Winn, “Optimal dispatch strategy in remote hybrid power systems”, *Sol. Energy*, c. 58, sy 4, ss. 165-179, Eki. 1996, doi: 10.1016/S0038-092X(96)00087-4.

[41] G. Tina, S. Gagliano, ve S. Raiti, “Hybrid solar/wind power system probabilistic modelling for long-term performance assessment”, *Sol. Energy*, c. 80, sy 5, ss. 578-588, May. 2006, doi: 10.1016/j.solener.2005.03.013.

[42] R. Dufo-López ve J. L. Bernal-Agustín, “Design and control strategies of PV-Diesel systems using genetic algorithms”, *Sol. Energy*, c. 79, sy 1, ss. 33-46, Tem. 2005, doi: 10.1016/j.solener.2004.10.004.

[43] A. Djellad, P.-O. Logerais, A. Omeiri, O. Riou, F. Delaleux, ve J. Durastanti, *Modelling of hybrid energy conversion system: wind turbine/photovoltaic source associated with battery/ultracapacitor storage*. 2014.

[44] P. Leader ve R. Lasseter, “Control and Design of Microgrid Components”, s. 257.

[45] H. Louie ve K. Strunz, “Superconducting Magnetic Energy Storage (SMES) for Energy Cache Control in Modular Distributed Hydrogen-Electric Energy Systems”, *Appl. Supercond. IEEE Trans. On*, c. 17, ss. 2361-2364, Tem. 2007, doi: 10.1109/TASC.2007.898490.

[46] A. M. O. Haruni ve M. ScEng, “A stand-alone hybrid power system with energy storage”, s. 198.

[47] L. Maharjan, S. Inoue, ve H. Akagi, “A Transformerless Energy Storage System Based on a Cascade Multilevel PWM Converter With Star Configuration”, *IEEE Trans. Ind. Appl.*, c. 44, sy 5, ss. 1621-1630, Eyl. 2008, doi: 10.1109/TIA.2008.2002180.

[48] J. Brownlee, “How to Reshape Input Data for Long Short-Term Memory Networks in Keras”, *Machine Learning Mastery*, 29 Ağustos 2017. https://machinelearningmastery.com/reshape-input-data-long-short-term-memory-networks-keras/ (erişim Oca. 09, 2022).

[49] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, ve S. Hochreiter, “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”, *ArXiv170608500 Cs Stat*, Oca. 2018, Erişim: 09 Ocak 2022. [Çevrimiçi]. Erişim adresi: http://arxiv.org/abs/1706.08500

[50] “An overview of gradient descent optimization algorithms”, *Sebastian Ruder*, 19 Ocak 2016. https://ruder.io/optimizing-gradient-descent/ (erişim Oca. 09, 2022).

[51] “Mean Squared Error (MSE)”, *Statistics By Jim*, 12 Kasım 2021. https://statisticsbyjim.com/regression/mean-squared-error-mse/ (erişim Oca. 09, 2022).

[52] “Mean Squared Error”, *The Science of Machine Learning*. https://www.ml-science.com/mean-squared-error (erişim Oca. 09, 2022).

[53] “Root-Mean-Squared Error - an overview | ScienceDirect Topics”. https://www.sciencedirect.com/topics/engineering/root-mean-squared-error (erişim Oca. 09, 2022).

[54] “Batch Size in a Neural Network explained”. https://deeplizard.com/learn/video/U4WB9p6ODjM (erişim Oca. 10, 2022).

[55] Hossain, Md. Alamgir, Ripon K. Chakrabortty, Sondoss Elsawah, Evan MacA. Gray, ve Michael J. Ryan. “Predicting Wind Power Generation Using Hybrid Deep Learning With Optimization”. *IEEE Transactions on Applied Superconductivity* 31, sy 8 (November 2021): 1-5.

[56] Hossain, Md Alamgir, Ripon K. Chakrabortty, Sondoss Elsawah, ve Michael J. Ryan. “Hybrid Deep Learning Model for Ultra-Short-Term Wind Power Forecasting”. İçinde *2020 IEEE International Conference on Applied Superconductivity and Electromagnetic Devices (ASEMD)*, 1-2, 2020.

[57] Hadayeghparast, Shahrzad, Amir Namavar Jahromi, ve Hadis Karimipour. “A Hybrid Deep Learning-Based Power System State Forecasting”. İçinde *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 893-98, 2020.