Jurnal Teknik Informatika (JUTIF)
Vol. 4, No. 6, Desember 2023, pp. 1347-1357

DOI: https://doi.org/10.52436/1.jutif.2023.4.6.1118
p-ISSN: 2723-3863
e-ISSN: 2723-3871

IMPLEMENTATION OF LSTM (LONG SHORT TERM MEMORY) ALGORITHM TO PREDICT WEATHER IN CENTRAL JAVA

Rhedy Irwan[*1], Pulung Nurtantio Andono[*2], Farrikh Al Zami[*3], Filmada Ocky Saputra[*4], Rama Aria Megantara[*5], L. Budi Handoko[*6], Chaerul Umam[*7]

[1,2,3,4,5,6,7]Study Program in Informatics Engineering, Faculty of Computer Science,
Universitas Dian Nuswantoro, Semarang, Indonesia
Email: [1]111201911946@mhs.dinus.ac.id, [2]pulung.nurtantio.andono@dsn.dinus.ac.id, [3]alzami@dsn.dinus.ac.id, [4]filmada.os@dsn.dinus.ac.id, [5]aria@dsn.dinus.ac.id, [6]handoko@dsn.dinus.ac.id, [7]chaerul@dsn.dinus.ac.id

(Article received: June 11, 2023; Revision: July 17, 2023; published: Desember 23, 2023)

Abstract

Agro-indutrial agricultural production such as red onions in Indonesia has a very important share in driving Indonesia's economic growth, especially in Central Java province which contributed 28.15% of the total national red onion production in 2021. Weather conditions have a major influence on the red onion planting process until the red onions are ready to be harvested. In this study, the objective is to predict various types of weather such as rainfall, air temperature, and air humidity in seven districts in Central Java, namely Brebes, Temanggung, Demak, Boyolali, Kendal, Pati, and Tegal. To do this, the use of the LSTM (Long Short Term Memory) algorithm with its ability to store memory longer than RNN will be reliable for predicting various types of weather in the future. This research was developed with the CRISP-DM (Cross Industry Process Model for Data Mining) method which has a goal-oriented approach, this method is a mature and widely accepted method in Data Mining with various applications in Machine Learning. With the final results from 39 models by using the evaluation of the average value of train MSE 0.013, test RMSE 0.11, test MSE of 0.02, test RMSE 0.12 and succeed to predict 5 days or months ahead from the last data that is provided.

Keywords: *Central Java, LSTM, MSE, RMSE, Weather Prediction*.

1. INTRODUCTION

In Indonesia, agro-industrial agricultural production has a contribution to GDP of 13.5% or an increase of 1% from 2019. So that agro-industrial agricultural products are from agriculture, fisheries/marine, livestock, farm, and forestry sectors are very important to encourage Indonesia's economic growth [1] [2]. Not only as a source of daily food for the community[3], but also as a source of state income. The agricultural sector remains a mainstay in absorbing labor from time to time [4] due to the habitual nature of its activities and the constant need for products from agriculture [5].

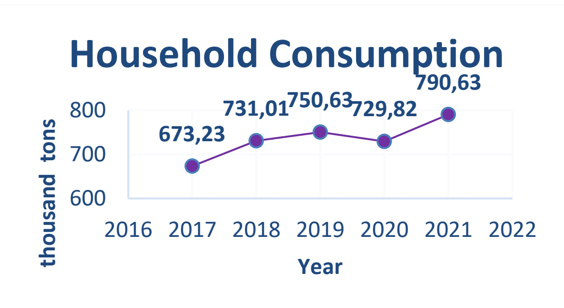


Figure 1. Red Onion Consumption by Indonesian Households

One of the high-value agricultural products in the Indonesian market is red onions [6]. Red onions (Allium cepa L.) [7]in addition to high value is one of the most important horticultural vegetables in the world [8]. Because it can be used in all aspects of life ranging from health to kitchen spices in processing food.

Based on the Central Bureau of Statistics (BPS), in 2021, red onion production managed to increase by 10.43% (189.15 kilotons) compared to 2020.

The household sector Figure 1 contributed 790.63 kilo tons of red onion consumption, an increase of 8.33% compared to 2020. For the entire household red onion consumption itself reaches 94.16% of the total existing red onion consumption.

In addition, according to the Central Statistics Agency (BPS), the results of red onion production activities in 2021 fluctuate, production in August is the highest reaching 218.74 kilo tons (See Figure 2). with a harvest area of 18,070 hectares. The provinces with the highest red onion production are Central Java, East Java, and West Nusa Tenggara. Central Java Province contributes as much as 28.15% of Indonesia's production which reaches 564.26 kilo tons and a harvest area of around 55.98 thousand hectares.

With the achievement of red onion production and consumption activities, there are many challenges, one of which is the weather [9]. Weather conditions such as temperature, rainfall, and air humidity can cause various diseases that can risk crop failure [10] .

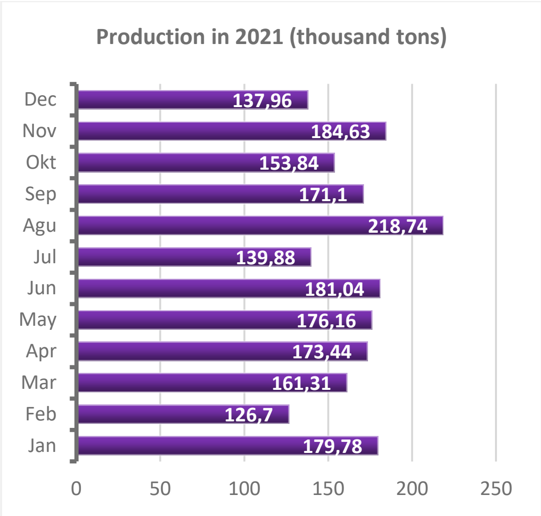


Figure 2. Red onion production per month in 2021 in Indonesia

For this reason, this study conducted a survey of seven districts including Brebes, Temanggung, Demak, Boyolali, Kendal, Pati, and Tegal in Central Java Province. By analyzing the weather using data (Google Earth Engine MAP) GEEMAP and (Meteorology, Climatology, and Geophysical Agency) BMKG to determine the level of humidity, temperature, and rainfall. So that we can make a prediction system to find out the weather in the future.

Judging from the existing data, it can be predicted how the weather in 7 districts using the LSTM (Long short-term memory) algorithm. The fundamental difference between LSTM and RNN is that LSTM compensates for the shortcomings of its predecessor, RNN (Recurrent Neural Network). RNNs cannot predict data based on information stored in the long term. In other words, the issue of storage duration is not an issue with LSTM [11]. Systems that implement LSTM can process, predict, and classify information based on time series data [12]. According to this concept, LSTM can recall and delete old data when it is no longer needed [13]. Therefore, information management becomes more complete and up-to-date [14]. As conducted by Chenjia Hu and his colleagues entitled "Prediction of ultra-short-term wind power based on CEEMDAN-LSTM-TCN" the LSTM algorithm has an error in MSE and RMSE smaller than other algorithms [15].

2. RESEARCH METHOD

This research method uses a combination of analysis with the CRISP-DM method which can be seen in Figure 3. The CRISP-DM (Cross Industry Process Model for Data Mining) methodology, which uses a goal-oriented approach, was used to structure this research. The CRISP-DM methodology is a mature methodology that is consistently well accepted in data mining research using machine learning. It offers a lifecycle approach to research involving applied artificial intelligence, and is considered the best methodology for knowledge discovery in databases (KDD) [16]. There are a number of features of CRISP-DM that make it suitable for evidence mining. In addition, this technique offers a general process model that summarizes the overall framework and aspects of the methodology where it then offers specialization based on a pre-defined context. Figure 3 details the many stages of the process and outcomes to validate the success of the prediction.

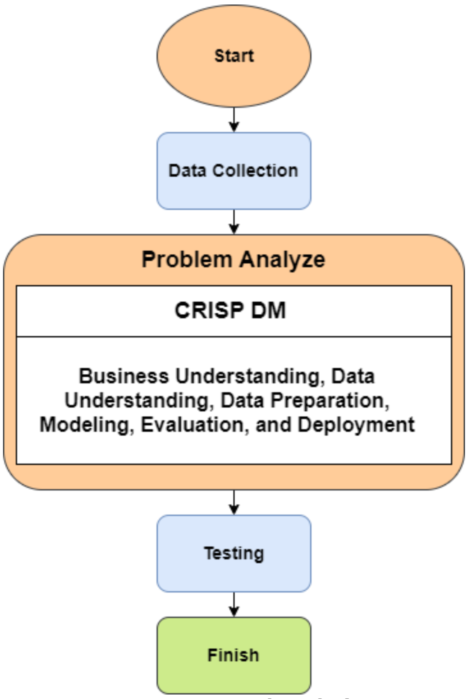


Figure 3. Research Method

CRISP-DM follows six main steps, namely Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment[16].

2.1. Business Understanding

Business Understanding is a step to understand the purpose of the research and how the research can help in making business decisions. This step also involves identifying the hypotheses and target variables to be analyzed.

2.2. Data Understanding

Data Understanding is the step to collect all the data required for the research. This can be internal company data or publicly available external data.

2.3. Data Preparation

Data preparation is a step to clean and prepare the data to make it ready for processing by the model. This can involve data grouping, removal of useless data, data measurement, and normalization. Python is used in this procedure since it has grown to be one of the most widely used platforms and the top open-source programming language for deep learning[17] Python on Google Collaboratory is an open and cloud-friendly notebook environment. This tool helps users and their team members to edit documents and supported libraries that are often used for research, especially when related to machine learning [18].

Normalization is required in order to be processed in machine learning. Normalization in this case uses a library from Python, namely Scikit-learn. A Python package called Scikit-learn integrates many types of cutting-edge computers to learn about the techniques for supervised and unsupervised situations. The library focuses on how machines acquire knowledge using a high-level, general-purpose language aimed at non-specialists[19]. Normalization by using function MinMaxScaler, Considering the weather is a time series of data in T times with interval X = X = [$x_1, x_2, \ldots, x_T$

$$x_1 = \frac{x_t - \bar{x}}{s_x} \quad (1)$$

$$x'_t = \frac{x_t - xmin}{xmax - xmin} \quad (2)$$

Where is the observation value at t, is the data that has been normalized at t [20]. After normalization, the data is displaced or separated into train data, and test data. The train data amounts to 80% of the total data and the remaining 20% is used for test data then make new data series so that can be processed in modelling stage.

2.4. Modeling

Modeling is the step to build a model that can be used to predict the target variable. This can involve selecting an appropriate modeling method and training the model using training data in Deep Learning. Deep learning is a subset of Machine Learning that is fueled by the massive rise in processing power in computers or machines. Deep learning algorithms are commonly used in pattern recognition systems due to their ability to extract abstract ideas from high-dimensional data [21]. Tensorflow, an open source deep learning platform for developers, is used to build Machine Learning and Deep Learning applications. To conceive and investigate intriguing ideas concerning Google's artificial intelligence. TensorFlow is written in the Python programming language, hence it is considered a simple framework [22].

LSTM is a standard variant of RNN. The standard RNN is quite simple and robust. However, in practice it is difficult to train the model for problems with a long time lag between the target and the previous related event. Hence LSTM was introduced to overcome the problems faced by RNNs[[20].

The architecture of the RNN algorithm is designed for transient sequence models. LSTM has a long-range dependency which makes LSTM more accurate than conventional RNNs. The backpropagation algorithm in RNN causes errors in its backflow problem, Unlike RNN, LSTM contains specialized units called memory blocks in the recurrent hidden layer. Memory blocks contain memory cells with self-connection to store the temporal state of specialized networks called gates that are useful for controlling the flow of information. Each memory block in the original architecture contains three types of gates viz: Input Gate: Input gates control the flow of input activation to the memory cells. Output Gate: Output gates control the flow of cell activation outputs to the rest of the network. Forget Gate: Scales the internal state of the cell before it is added as an input to the memory cell through self-recurrent on the cell connection, therefore adaptively forgetting or resetting the memory cell [20].

In addition, modern LSTM algorithms contain internal cells in their gates that are used to learn how to properly time the output. To simplify analysis, the LSTM architecture is often used in the tt(time) dimension as in the following diagram (See Figure 4)

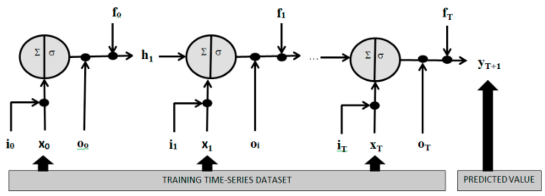


Figure 4. Arsitektur of Folded Long Short-Term Memory Model

In the diagram above Figure 4, it can be seen that each LSTM block receives signals from: input signal (x), input gate signal (i), recurrent signal (h), forget gate signal (f), and produces output gate signal (o). The process flow in each LSTM memory block can be depicted in the diagram below (See Figure 5).

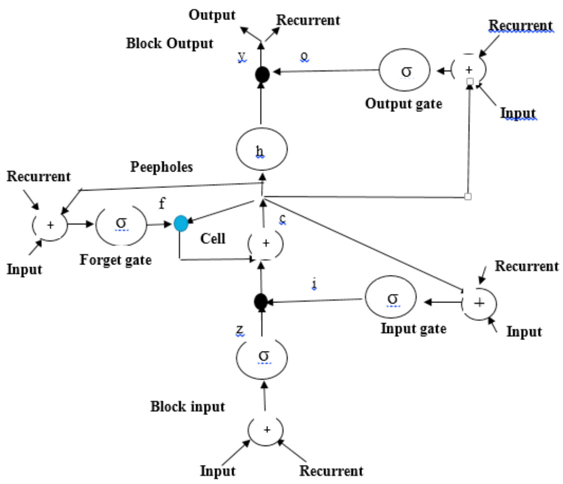


Figure 5. Arsitektur of Folded Long Short-Term Memory Model

An LSTM network computes a mapping from the sequence of inputs x = (x1, …,xt) to the sequence of outputs y = (y1, ..., yt) by computing the network's activation units with a recurring equation from t = 1 to T as shown below.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (3)$$

$$z_t = tanh(W_z x_t + U_z h_{t-1} + b_z) \quad (4)$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_i\right) \quad (5)$$

$$C_t = i_t * z_t + f_t * C_{t-1} \quad (6)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o C_t + b_o) \quad (7)$$

$$h_t = o_t * tanh(C_t) \quad (8)$$

WhereW_i,W_z,W_f,W_o,U_i,U_z, U_f, U_o are the model parameters that are performed while the model is trained; (sigmoid) and is the activation function and is the bias. The LSTM layer can be used with tensorflow and Keras library from python. Keras follows best practices to reduce cognitive load in a consistent and simple manner. Keras minimizes the number of user actions required for common use cases, and provides clear and actionable feedback on user errors. This makes Keras easy to learn and use. Easy usage does not reduce flexibility as Keras can be integrated with low-level Deep Learning languages such as TensorFlow. This makes it possible to implement anything that can be created in that base language [23].

The dropout technique is an effective regularization method to reduce overfitting problems. The core idea of dropout is to prevent the network from relying too much on each neuron and thus reducing the adaptability between neurons. The neurons are multiplied by a random variable that has a probability p and follows a Bernoulli distribution throughout the training phase at each iteration, according to the mechanism. The dropout rate corresponds to Figure 6 which illustrates the difference in structure between models with and without dropout. The corresponding formula without dropout is as follows [24].

$$\tilde{p} = w_o h_t + b_o \quad (9)$$

With *dropout*:

$$\tilde{p}_t = w_o \tilde{h}_t + b_o = w_o(r_t \odot h_t) + b_o \quad (10)$$

Where is the output of the model at time t before being processed by its active function. represents the output vector of the hidden layer and represents the weight matrix and can that connects the hidden layer and output layer [19]. The final output is.

$$p_t = f(\tilde{p}_t) \quad (11)$$

Where is the output of the model and is the activation function of the output layer [19].

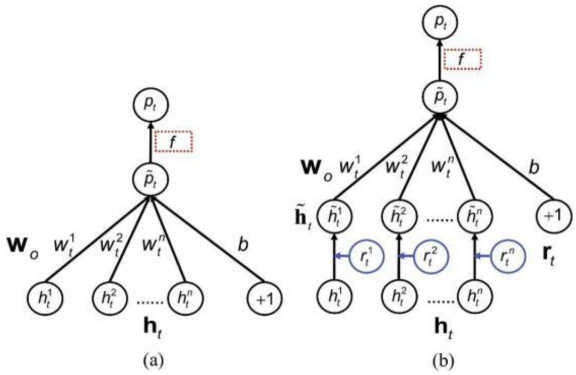


Figure 6. Dropout technique for overfitting: (a) standard network; (b) network with dropout

2.5. Evaluation

Evaluation is the step to evaluate the ability of the model to predict the target variable. This can involve using test data to obtain the model error score and comparing it with other methods that may be used.

In the evaluation step, it can be seen from the error value or how the model fitting graph runs well or not. To check, you can use a library from python, namely Matplotlib. Matplotlib is a data visualization package that is widely used in Python. Matplotlib can easily draw a variety of high-quality 2D graphs as well as some pretty simple 3D graphs. Matplotlib, a Python library, provides a simple language, good drawing accuracy, and simple and easy-to-understand code. [25]. Alternatively, it can use the seaborn library which has functions for dataset-oriented visualization, making it easier to translate questions about data into graphs. Seaborn is designed to be useful throughout the lifecycle of scientific research [26].

RMSE stands for Root Mean Squared Error. It is a measure of the difference between the value predicted by the model or estimator and the true value. RMSE is a popular measure of accuracy for continuous data, and is a commonly used metric in the field of machine learning. The RMSE equation is calculated as the square root of the average squared difference between the predicted value and the true value. In mathematical notation, this is represented as:

$$RMSE = \sqrt{\sum_{i-1}^{n}\left(x_{predicted}[i] - x_{actual}[i]\right)^2} \quad (12)$$

The smaller the RMSE value, the better the model predicts the true value. Where: n is the number of samples, $x_{predicted}[i]$ is the predicted value for the i-th sample, $x_{actual}[i]$ is the true value for the i-th sample[27].

2.6. Deployment

Deployment is the step to generate the model into files that can be used for various platforms. Overall, CRISP-DM is a useful methodology for extracting information from data and making business decisions. It provides a clear structure to guide the process of extracting information from data and ensures that research can be completed effectively and efficiently.

3. RESULT AND DISCUSSION

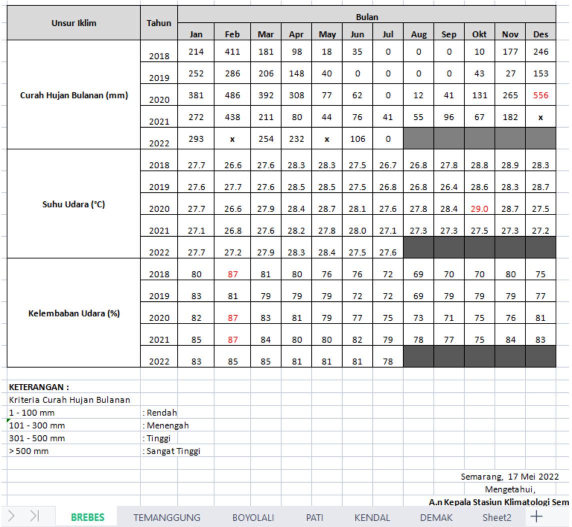
3.1. Business Understanding

Ideal weather is needed to get good red onion production. If the weather can be predicted, it is expected to increase production or reduce losses in the planting process, for example, what red onion farmers really want to avoid is crop failure.

Modeling predictions for various weather factors, such as temperature, rainfall, ground surface temperature and others are needed to solve these business problems by looking at the model's ability to evaluate the minimum possible error.

3.2. Data Understanding

In the data obtained there are two different sources, namely from BMKG (Meteorology, Climatology, and Geophysical Agency) and GEE (Google Earth Engine). BMKG data in the form of monthly global rainfall data from 6 districts namely Brebes, Temanggung, Boyolali, Pati, Kendal, and Demak in 2018 to 2022 in Central Java. It can be seen in Figure 7 that the data arrangement is not neat and there is a lot of graphic content such as colors and logo images that are not needed. Data processing in machine learning requires clean and tidy data. To tidy up the untidy table, you can use the melt function in python which is already contained in the python library. Then for the x value which is empty data, it can be replaced with the mean or average of the number of rows where the empty data is located. GEEMAP data is daily global data of various weather such as rainfall, soil temperature, air temperature, and land surface temperature from 2018 to August 2022 from 7 districts of Brebes, Tegal, Temanggung, Boyolali, Pati, Kendal, and Demak in Central Java Province.



Unsur Iklim	Tahun	Bulan											
		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Okt	Nov	Des
Curah Hujan Bulanan (mm)	2018	214	411	181	98	18	35	0	0	0	10	177	246
	2019	252	286	206	148	40	0	0	0	0	43	27	153
	2020	381	486	392	308	77	62	0	12	41	131	265	556
	2021	272	438	211	80	44	76	41	55	96	67	182	x
	2022	293	x	254	232	x	106	0					
Suhu Udara (°C)	2018	27.7	26.6	27.6	28.3	28.3	27.5	26.7	26.8	27.8	28.8	28.9	28.3
	2019	27.6	27.7	27.6	28.5	28.5	27.5	26.8	26.8	26.4	28.6	28.3	28.7
	2020	27.7	26.6	27.9	28.4	28.7	28.1	27.6	27.8	28.4	29.0	28.7	27.5
	2021	27.1	26.8	27.6	28.2	27.8	28.0	27.1	27.3	27.3	27.5	27.3	27.2
	2022	27.7	27.2	27.9	28.3	28.4	27.5	27.6					
Kelembaban Udara (%)	2018	80	87	81	80	76	76	72	69	70	70	80	75
	2019	83	81	79	79	79	72	72	69	79	79	79	77
	2020	82	87	83	81	79	77	75	73	71	75	76	81
	2021	85	87	84	80	80	82	79	78	77	75	84	83
	2022	83	85	85	81	81	81	78					

KETERANGAN :
Kriteria Curah Hujan Bulanan
1 - 100 mm : Rendah
101 - 300 mm : Menengah
301 - 500 mm : Tinggi
> 500 mm : Sangat Tinggi

Semarang, 17 Mei 2022
Mengetahui,
A.n Kepala Stasiun Klimatologi Sem

BREBES TEMANGGUNG BOYOLALI PATI KENDAL DEMAK Sheet2

Figure 7. Monthly weather in six regencies of Central Java

3.3. Data Preparation

In the data preparation step by selecting the column to be processed. In this case, we use the Demak Rain Monthly column. If visualized using the matplotlib library, it will look like Figure 8.

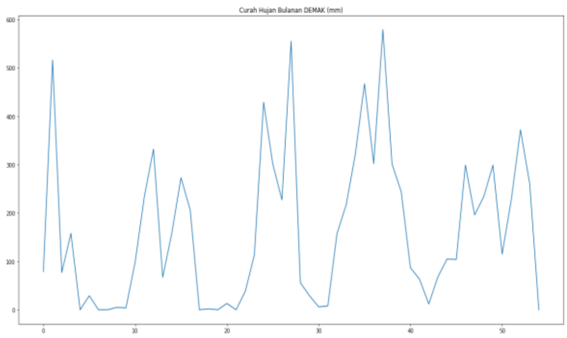


Figure 8. Visualization Monthly Rainfal in Demak regency year of 2018-2022

Visualization in Figure 8 can explain that rainfall in the selected column, namely in Demak district, has a seasonal pattern.

Furthermore, it will require normalization to be processed in machine learning. Normalization this time uses the MinMaxScaler function in the sklearn library with its feature range between values 0 to 1 and reshape between -1 and 1. After normalization, the data is split into train data and test data. The train data amounts to 80% of the total data and the remaining 20% is used for test data.

The new data series is used to create a new dataset into the time series to be processed in the LSTM algorithm. This new data series has two parameters, namely dataset and step. Step here is a time_stamp variable which is worth 3. The step parameter is a step in each time series process. For example, there is a sequence of values in the dataset such as X[100, 110, 120] = Y[130], X[110,120,130] = Y[140]. X is the train or test data and Y is the data to be predicted. The values 100, 110, 120 are time_stamps that are worth 3 steps. Then the value 130 in Y[130] will go down to X, namely X[110,120,130] for the next process until the last data.

3.4. Modeling

The model is built using the Tensorflow framework using the Long Short-Term Memory algorithm or commonly referred to as LSTM as its main architecture. Tuning hyperparameters in the LSTM model there are 10 layers where 4 layers with the LSTM algorithm with each nodes of 32, 64, 128, 256, dropout which is useful for the model to fit smoothly so that overfitting or underfitting can be avoided. Each Dropout is worth 0.2 or 20%, 1 layer for Dense with 1 node see Figure 9.

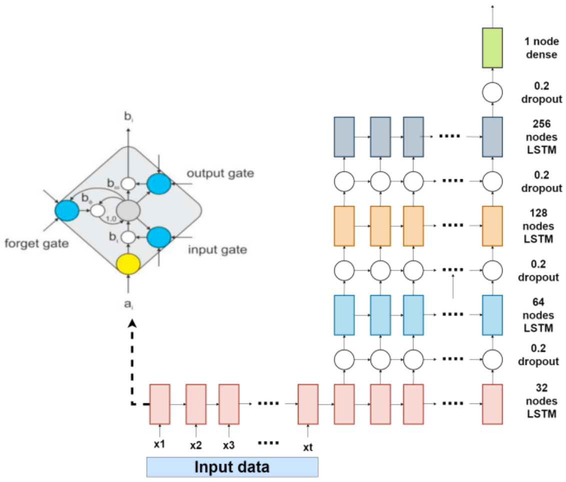


Figure 9. Research Model Framework

Figure 10 demonstrates the building of an LSTM memory cell, which is a fundamental unit of the LSTM model. As previously stated, each memory cell has an input gate that learns to protect the memory cell's continuous error flow against irrelevant inputs. The output gate unit learns to protect other units from the memory cell's irrelevant memory contents. The forget gate unit learns to regulate how long a value remains in the memory cell. In this case, the input data consists of predicted variable weather data that has been chosen. The projected variable data visibility is the output.

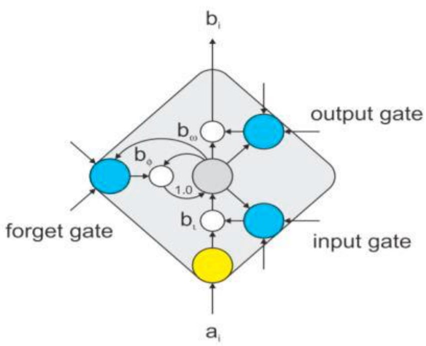


Figure 10. Structure of LSTM cell

3.5. Evaluation

Evaluation of model performance can be done by looking at the visualization of the fitting process when modeled. By using the matplotlib library can visualize the results of a model's performance.

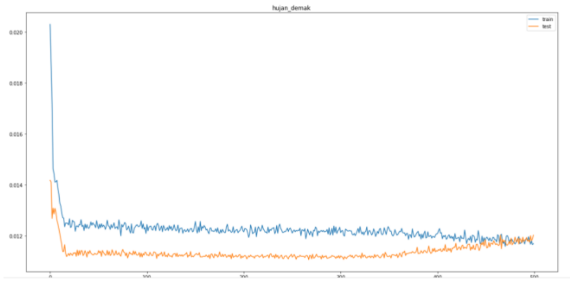


Figure 11. Loss MSE for Monthly Rainfall Demak

Figure 11 explains that the loss performance where using MSE or Mean Squared Error can work well between Train and Test results. Fitting is found when epochs are close to the value of 500.

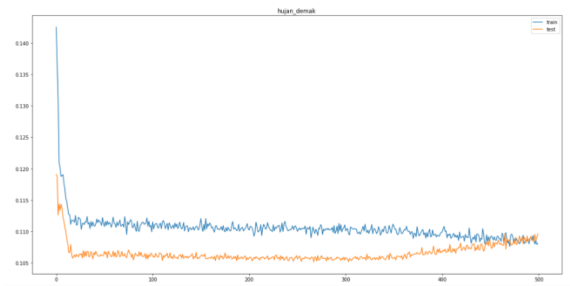


Figure 12. Loss RMSE for Monthly Rainfall Demak

Likewise, in performance metrics that use RMSE or Root Mean Squared Error. Fitting runs well and meets at epochs around almost 500 can be seen in Figure 12. After selecting each column where for modeling which consists of BMKG (Meteorology, Climatology, and Geophysical Agency) data consisting of Demak, Boyolali, Pati, Temanggung, Brebes, Kendal districts with weather such as Rainfall, Air Temperature, and Air Humidity where monthly data from January 2018 to August 2022. Then data from GEE (Google Earth Engine) which has daily weather data such as rainfall, soil temperature, and air temperature from 2018 to August 2022 from 7 districts of Brebes, Tegal, Temanggung, Boyolali, Pati, Kendal, and Demak. Model error assessment uses MSE, and RMSE with the help of the scikit-learn library. The results of models can be checked Table 1 and Table 2 with evaluation of the average value of the entire model are error train MSE 0.013 and RMSE 0.11 for the average error test MSE of 0.02 and RMSE 0.12.

Table 1. Results of Train MSE and Train RMSE each model

Weather	Train MSE	Train RMSE
Daily Demak Rainfall	0.011485593	0.107170857
Daily Demak Air Temperature	0.008942584	0.094565235

Daily Demak Soil Temperature	0.006838103	0.082692824
Daily Boyolali Rainfall	0.017478809	0.132207453
Daily Boyolali Air Temperature	0.009828976	0.099141195
Daily Boyolali Soil Temperature	0.00610309	0.078122281
Daily Pati Rainfall	0.016190888	0.127243415
Daily Pati Air Temperature	0.011401984	0.106780075
Daily Pati Soil Temperature	0.004726454	0.068749212
Daily Temanggung Rainfall	0.0130376	0.114182308
Daily Temanggung Air Temperature	0.007623303	0.087311529
Daily Temanggung Soil Temperature	0.006877598	0.082931288
Daily Brebes Rainfall	0.016769264	0.129496187
Daily Brebes Air Temperature	0.009032979	0.095041983
Daily Brebes Soil Temperature	0.004076999	0.063851386
Daily Kendal Rainfall	0.017911386	0.133833423
Daily Kendal Air Temperature	0.008263543	0.090904035
Daily Kendal Soil Temperature	0.005103064	0.071435735
Daily Tegal Rainfall	0.013961938	0.118160643
Daily Tegal Air Temperature	0.009072894	0.095251739
Daily Tegal Soil Temperature	0.004641352	0.068127468
Monthly Demak Rainfall	0.019216761	0.138624534
Monthly Demak Air Temperature	0.016723309	0.12931864
Monthly Demak Air Humidity	0.010128999	0.100642927
Monthly Boyolali Rainfall	0.011211301	0.105883434
Monthly Boyolali Air Temperature	0.019185115	0.138510346
Monthly Boyolali Air Humidity	0.025213098	0.158786327
Monthly Pati Rainfall	0.019739538	0.140497461
Monthly Pati Air Temperature	0.016992837	0.13035658
Monthly Pati Air Humidity	0.010250747	0.101245977
Monthly Temanggung Rainfall	0.016130451	0.127005711
Monthly Temanggung Air Temperature	0.009551488	0.097731717
Monthly Temanggung Air Humidity	0.038542956	0.196323603
Monthly Brebes Rainfall	0.009784665	0.098917462
Monthly Brebes Air Temperature	0.035992481	0.189716846
Monthly Brebes Air Humidity	0.01577029	0.125579819
Monthly Kendal Rainfall	0.004999501	0.07070715
Monthly Kendal Air Temperature	0.017092988	0.130740151
Monthly Kendal Air Humidity	0.010832237	0.104078032
Average	0.013249414	0.111073512

Table 2. Results of Test MSE, Test RMSE, and Fitting each model

Weather	Test MSE	Test RMSE	Fitting
Daily Demak Rainfall	0.011759466	0.10844107	Fitting
Daily Demak Air Temperature	0.010377092	0.101868011	Overfitting
Daily Demak Soil Temperature	0.004093772	0.063982591	Underfitting
Daily Boyolali Rainfall	0.026501842	0.16279386	Underfitting
Daily Boyolali Air Temperature	0.007638756	0.087399974	Underfitting
Daily Boyolali Soil Temperature	0.002320618	0.048172791	Underfitting
Daily Pati Rainfall	0.017336376	0.131667674	Overfitting
Daily Pati Air Temperature	0.014267963	0.11944858	Overfitting
Daily Pati Soil Temperature	0.001639645	0.040492535	Underfitting
Daily Temanggung Rainfall	0.016823087	0.12970385	Underfitting
Daily Temanggung Air Temperature	0.005472283	0.073974878	Underfitting
Daily Temanggung Soil Temperature	0.00275075	0.052447591	Underfitting
Daily Brebes Rainfall	0.021967486	0.148214325	Underfitting
Daily Brebes Air Temperature	0.010854711	0.104185946	Overfitting
Daily Brebes Soil Temperature	0.001113922	0.033375472	Underfitting
Daily Kendal Rainfall	0.021512926	0.14667286	Underfitting
Daily Kendal Air Temperature	0.011120962	0.105455972	Overfitting
Daily Kendal Soil Temperature	0.001078514	0.03284074	Underfitting
Daily Tegal Rainfall	0.018639838	0.136527792	Underfitting
Daily Tegal Air Temperature	0.01094312	0.10460937	Overfitting
Daily Tegal Soil Temperature	0.001364287	0.036936261	Underfitting
Monthly Demak Rainfall	0.026429284	0.162570864	Fitting
Monthly Demak Air Temperature	0.031571381	0.177683368	Overfitting
Monthly Demak Air Humidity	0.004173623	0.064603582	Underfitting
Monthly Boyolali Rainfall	0.061255459	0.247498408	Overfitting
Monthly Boyolali Air Temperature	0.02522954	0.158838093	Overfitting
Monthly Boyolali Air Humidity	0.013609034	0.116657764	Underfitting
Monthly Pati Rainfall	0.05609389	0.236841485	Overfitting
Monthly Pati Air Temperature	0.030594107	0.174911708	Overfitting
Monthly Pati Air Humidity	0.004353273	0.065979339	Underfitting
Monthly Temanggung Rainfall	0.071020864	0.266497403	Overfitting
Monthly Temanggung	0.057857245	0.240535334	Overfitting

Air Temperature			
Monthly Temanggung Air Humidity	0.024181632	0.15550445	Underfitting
Monthly Brebes Rainfall	0.044963319	0.212045565	Overfitting
Monthly Brebes Air Temperature	0.030137852	0.173602566	Fitting
Monthly Brebes Air Humidity	0.016068008	0.126759648	Fitting
Monthly Kendal Rainfall	0.031819426	0.178379998	Overfitting
Monthly Kendal Air Temperature	0.029047927	0.17043452	Overfitting
Monthly Kendal Air Humidity	0.00636202	0.079762273	Fitting
Average	0.020111418	0.127649193	

After going through the fitting process, the model can predict future weather, here trying to predict the next 5 days with daily weather data in Table 3. D-day means the last day from existing data.

Table 3. Prediction daily weather 5 days ahead after D-day

Weather	D+1	D+2	D+3	D+4	D+5
Daily Boyolali Air Temperature	23.51	23.55	23.63	23.68	23.73
Daily Boyolali Rainfall	8.47	21.77	18.60	15.95	14.81
Daily Boyolali Soil Temperature	15.30	15.30	15.30	15.29	15.29
Daily Brebes Air Temperature	27.72	27.69	27.70	27.68	27.66
Daily Brebes Rainfall	5.49	6.52	9.24	8.29	8.74
Daily Brebes Soil Temperature	15.42	15.42	15.41	15.41	15.41
Daily Demak Air Temperature	28.61	28.60	28.58	28.57	28.56
Daily Demak Rainfall	6.58	7.05	7.15	7.18	7.28
Daily Demak Soil Temperature	15.33	15.39	15.36	15.38	15.37
Daily Kendal Air Temperature	27.25	27.25	27.24	27.22	27.20
Daily Kendal Rainfall	1.66	2.83	3.99	5.28	6.35
Daily Kendal Soil Temperature	15.36	15.36	15.36	15.35	15.35
Daily Pati Air Temperature	26.97	27.01	26.99	26.98	26.97
Daily Pati Rainfall	8.08	8.41	7.88	7.10	6.85
Daily Pati Soil Temperature	15.45	15.46	15.46	15.46	15.46
Daily Tegal Air Temperature	27.69	27.65	27.66	27.62	27.59
Daily Tegal Rainfall	4.25	5.57	7.81	6.93	7.54
Daily Tegal Soil Temperature	15.46	15.45	15.45	15.45	15.45
Daily Temanggung Air Temperature	22.20	22.30	22.38	22.45	22.50
Daily Temanggung Rainfall	2.23	3.55	4.33	6.21	7.75
Daily Temanggung Soil Temperature	15.25	15.27	15.27	15.27	15.27

In Figure 13 the orange color explains the graph of how the weather prediction is 5 months ahead after the last data, which is in blue from Monthly demak rainfall.

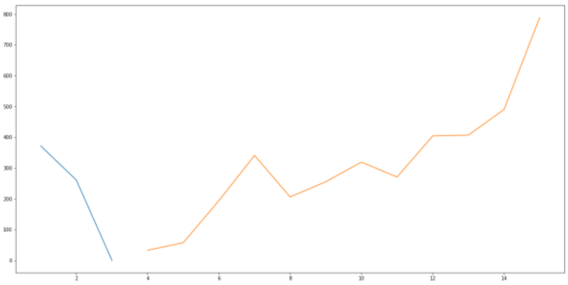


Figure 13: Prediction 5 months ahead from monthly demak rainfall (orange)

Then in table 14 is the result of predicting the next 5 months after the last month of existing data. When viewed in Figure 7 the last month is July in 2022 which explains as M-Month.

Table 4. Prediction monthly weather 5 months ahead after M-month

Weather	M+1	M+2	M+3	M+4	M+5
Monthly Boyolali Air Humidity	76.72	76.54	80.03	80.72	81.74
Monthly Boyolali Air Temperature	26.90	27.10	27.20	27.00	26.96
Monthly Boyolali Rainfall	68.66	-1.22	260.70	147.81	682.92
Monthly Brebes Air Humidity	77.27	76.35	75.36	74.49	74.11
Monthly Brebes Air Temperature	27.53	27.78	27.86	27.87	27.75
Monthly Brebes Rainfall	-11.07	-24.25	-6.94	18.54	67.58
Monthly Demak Air Humidity	73.25	71.04	70.64	72.18	75.63
Monthly Demak Air Temperature	28.34	28.53	28.49	28.32	28.17
Monthly Demak Rainfall	33.07	57.72	194.21	340.98	206.29

Monthly Kendal Air Humidity	72.71	69.98	69.04	70.40	74.15
Monthly Kendal Air Temperature	28.29	28.47	28.42	28.27	28.14
Monthly Kendal Rainfall	0.55	7.59	79.03	311.37	489.85
Monthly Pati Air Humiditiy	73.21	71.11	70.77	72.28	75.52
Monthly Pati Air Temperature	28.36	28.53	28.48	28.34	28.23
Monthly Pati Ranfall	6.10	32.45	51.50	76.01	101.26
Monthly Temanggung Air Humdity	82.18	82.05	82.14	82.15	82.16
Monthly Temanggung Aur Temperature	23.12	23.18	23.38	23.73	23.95
Monthly Temanggung Rainfall	6.42	-24.05	65.10	126.53	298.47

By combining existing data and predicted data for the next few months. The visualization can be seen in Figure 14 below.

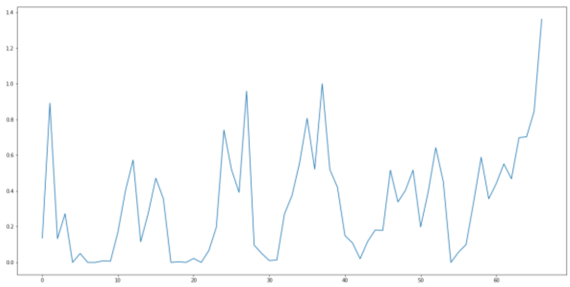


Figure 14: Combining existing data and future data from monthly demak rainfall

3.6. Deployment

In this deployment process by producing models into files that can be used for various platforms such as tflite for use in android applications, and json for website development.

4. DISCUSSION

Research using the LSTM algorithm has been widely used, such as the one conducted by Manzhu Yu et. al entitled "Using long short-term memory (LSTM) and Internet of Things (IoT) for localized surface temperature forecasting in an urban environment in 2021". This research shows that LSTM outperforms traditional time series forecasting techniques with RMSE values for minimum, average and maximum of 2.71/2.99/3.31 [28]. Then the research of Alfan Galih Salman and his friends about "Single Layer & Multi-layer Long Short-Term Memory (LSTM) Model with Intermediate Variables for Weather Forecasting in 2018". Comparing single layer and multi layer LSTM models using weather datasets (temperature, pressure, humidity, and dew point). The results of his research produced the best results on the weather variable pressure with an RMSE value of 0.0775 [20]. Furthermore, there is research on "Deep learning model for daily rainfall prediction: case study of Jimma, Ethiopia in 2022" by Demeke Endalie et al[29]. Rainfall prediction is an important task for some people, especially in the agricultural sector. This research was conducted in Jimma one of the regions in Ethiopia using LSTM algorithm which produces the best performance of RMSE of 0.01 compared to other algorithms such as kNN, SVM, and Decision Tree.

Based on researches above, LSTM algorithm has best performance than the other algorithms. Because of that, the author is interested in knowing how to implement the LSTM algorithm, especially to predict various weather in districts of Central Java using the latest dataset from 2018 to 2022. Comparing from those researches results, error found out the final results from 39 models by using the evaluation of the average value of train MSE 0.013, test RMSE 0.11, test MSE of 0.02, test RMSE 0.12.

Each column requires different hyperparameter tuning in modeling. This is needed to avoid overfitting or underfitting. Some models for districts are still underfitting or overfitting. It is necessary to change the hyperparameter tuning, the number of epochs, the value of nodes in each layer, fill missing values which can use mode, median or knn. By using class callbacks to avoid overfitting, or change the value of time_stamp in forming new time series data.

5. CONCLUSION

Based on the research conducted, from 39 total models. There are 5 fitting models, 16 overfitting models, and 18 underfitting models. Tuning the hyperparameters is recommended not to shuffle the data with Boolean True, time series data requires sequential data so the required status is False. Evaluation of the average value of the entire model error train MSE 0.013 and RMSE 0.11 for the average error test MSE of 0.02 and RMSE 0.12 .Models that work well when viewed from the fitting performance are Daily Demak Rainfall, Monthly Brebes Air Temperature, Monthly Brebes Air Humidity, Monthly Demak Rainfall, and Monthly Kendal Air Humidity.

ACKNOWLEDGMENT

We sincerely thank Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi, Republik Indonesia for partially funding this project through the Kedaireka Program. This work is also supported by Universitas Dian Nuswantoro (UDINUS) through the Center of Excellence in Science and Technology, UDINUS and RAMANI B.V. with the grant document contract:

Supply Chain and Customer Relationship Management Alignment on Red Onion Commodity using Artificial Inteligent based on Internet of Things and Blockchain, No. 176/E1/KS.06.02/2022.

REFERENCES

[1] P. N. Andono, F. O. Saputra, G. F. Shidik, and Z. A. Hasibuan, "End-to-End Circular Economy in Onion Farming with the Application of Artificial Intelligence and Internet of Things," in *2022 International Seminar on Application for Technology of Information and Communication: Technology 4.0 for Smart Ecosystem: A New Way of Doing Digital Business, iSemantic 2022*, 2022. doi: 10.1109/iSemantic55962.2022.9920447.

[2] Y. H. Asnawi *et al.*, "Engineering and Technology Challenges of the Energy Usage: An Estimation for Agro-Industrial in Indonesia," in *IOP Conference Series: Materials Science and Engineering*, 2020. doi: 10.1088/1757-899X/874/1/012034.

[3] F. Kharianti and N. A. B. Rahman, "An Analysis Behaviour Of Palm Sugar Traders In Mandailing Natal Based On Islamic Business Ethics Perspective," *J. Indones. Manag. JIM*, vol. 2, no. 2, 2022, doi: 10.53697/jim.v2i2.615.

[4] W. Sepriani and Yuliawati, "PENYERAPAN TENAGA KERJA OLEH SEKTOR PERTANIAN Tahun 2016-2021," *J. Samudra Ekon.*, vol. 6, no. 1, 2022.

[5] S. I. Kusumaningrum, "Pemanfaatan Sektor Pertanian Sebagai Penunjang Pertumbuhan Perekonomian Indonesia," *Transaksi*, vol. 11, no. 1, pp. 80–89, 2019.

[6] Y. R. Kang *et al.*, "Calorie restriction effect of heat-processed onion extract (ONI) using in vitro and in vivo animal models," *Int. J. Mol. Sci.*, vol. 19, no. 3, 2018, doi: 10.3390/ijms19030874.

[7] L. Ricciardi *et al.*, "Assessment of genetic diversity of the 'acquaviva red onion' (Allium cepa L.) apulian landrace," *Plants*, vol. 9, no. 2, 2020, doi: 10.3390/plants9020260.

[8] E. R. Araújo, D. P. Alves, and J. R. Knoth, "Weather-based decision support reduces the fungicide spraying to control onion downy mildew," *Crop Prot.*, vol. 92, pp. 89–92, 2017, doi: https://doi.org/10.1016/j.cropro.2016.10.022

[9] P. B. Leelavinodhan, M. Vecchio, F. Antonelli, A. Maestrini, and D. Brunelli, "Design and implementation of an energy-efficient weather station for wind data collection†," *Sensors*, vol. 21, no. 11, 2021, doi: 10.3390/s21113831.

[10] M. Isip, R. Alberto, and A. Biagtan, "Forecasting anthracnose-twister disease using weather based parameters: geographically weighted regression focus," *Spat. Inf. Res.*, vol. 29, no. 5, 2021, doi: 10.1007/s41324-021-00386-6

[11] N. K. Manaswi, *Deep Learning with Applications Using Python*. 2018. doi: 10.1007/978-1-4842-3516-4.

[12] B. Kamala and B. Latha, "Process Mining and Deep Neural Network approach for the Prediction of Business Process Outcome," in *2022 International Conference on Communication, Computing and Internet of Things, IC3IoT 2022 - Proceedings*, 2022. doi: 10.1109/IC3IOT53935.2022.9767941.

[13] E. Ivan and H. D. Purnomo, "FORECASTING PRICES OF FERTILIZER RAW MATERIALS USING LONG SHORT TERM MEMORY," *J. Tek. Inform. Jutif*, vol. 3, no. 6, pp. 1663–1673, Dec. 2022, doi: 10.20884/1.jutif.2022.3.6.433

[14] H. Chung and K. S. Shin, "Genetic algorithm-optimized long short-term memory network for stock market prediction," *Sustain. Switz.*, vol. 10, no. 10, 2018, doi: 10.3390/su10103765.

[15] C. Hu, Y. Zhao, H. Jiang, M. Jiang, F. You, and Q. Liu, "Prediction of ultra-short-term wind power based on CEEMDAN-LSTM-TCN," *Energy Rep.*, vol. 8, pp. 483–492, 2022, doi: https://doi.org/10.1016/j.egyr.2022.09.171.

[16] J. A. Solano, D. J. L. Cuesta, S. F. U. Ibáñez, and J. R. Coronado-Hernández, "Predictive models assessment based on CRISP-DM methodology for students performance in Colombia - Saber 11 Test," in *Procedia Computer Science*, 2021. doi: 10.1016/j.procs.2021.12.278.

[17] B. Zohuri, F. Mossavar-Rahmani, and F. Behgounia, "Chapter 26 - Python programming–driven artificial intelligence," in *Knowledge is Power in Four Dimensions: Models to Forecast Future Paradigm*, B. Zohuri, F. Mossavar-Rahmani, and F. Behgounia, Eds., Academic Press, 2022, pp. 827–836. doi: https://doi.org/10.1016/B978-0-323-95112-8.00026-X.

[18] B. Prashanth, M. Mendu, and R. Thallapalli, "Cloud based Machine learning with advanced predictive Analytics using Google Colaboratory," *Mater. Today Proc.*, 2021, doi: https://doi.org/10.1016/j.matpr.2021.01.800.

[19] P. Deekshith and R. P. Singh, "Review on

Advanced Machine Learning Model: Scikit-Learn," *Int. J. Sci. Res. Eng. Dev.*, vol. 3, no. 4, 2020.

[20] A. G. Salman, Y. Heryadi, E. Abdurahman, and W. Suparta, "Single Layer & Multi-layer Long Short-Term Memory (LSTM) Model with Intermediate Variables for Weather Forecasting," *Procedia Comput. Sci.*, vol. 135, pp. 89–98, 2018, doi: https://doi.org/10.1016/j.procs.2018.08.153.

[21] R. A. de Oliveira and M. H. J. Bollen, "Deep learning for power quality," *Electr. Power Syst. Res.*, vol. 214, p. 108887, 2023, doi: https://doi.org/10.1016/j.epsr.2022.108887.

[22] T. Point, "TensorFlow Tutorial," *Tutor. Point Pvt Ltd*, 2019.

[23] C. François, "Why use Keras - Keras Documentation," *Keras.Io*, 2018, [Online]. Available: https://keras.io/why-use-keras

[24] X. Wei, L. Zhang, H.-Q. Yang, L. Zhang, and Y.-P. Yao, "Machine learning for pore-water pressure time-series prediction: Application of recurrent neural networks," *Geosci. Front.*, vol. 12, no. 1, pp. 453–467, 2021, doi: https://doi.org/10.1016/j.gsf.2020.04.011.

[25] S. Cao, Y. Zeng, S. Yang, and S. Cao, "Research on Python Data Visualization Technology," in *Journal of Physics: Conference Series*, 2021. doi: 10.1088/1742-6596/1757/1/012122.

[26] M. Waskom, "seaborn: statistical data visualization," *J. Open Source Softw.*, vol. 6, no. 60, 2021, doi: 10.21105/joss.03021.

[27] H. Chen and X. Chang, "Photovoltaic power prediction of LSTM model based on Pearson feature selection," *Energy Rep.*, vol. 7, pp. 1047–1054, 2021, doi: https://doi.org/10.1016/j.egyr.2021.09.167.

[28] M. Yu, F. Xu, W. Hu, J. Sun, and G. Cervone, "Using Long Short-Term Memory (LSTM) and Internet of Things (IoT) for Localized Surface Temperature Forecasting in an Urban Environment," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3116809.

[29] D. Endalie, G. Haile, and W. Taye, "Deep learning model for daily rainfall prediction: case study of Jimma, Ethiopia," *Water Supply*, vol. 22, no. 3, pp. 3448–3461, Nov. 2021, doi: 10.2166/ws.2021.391.