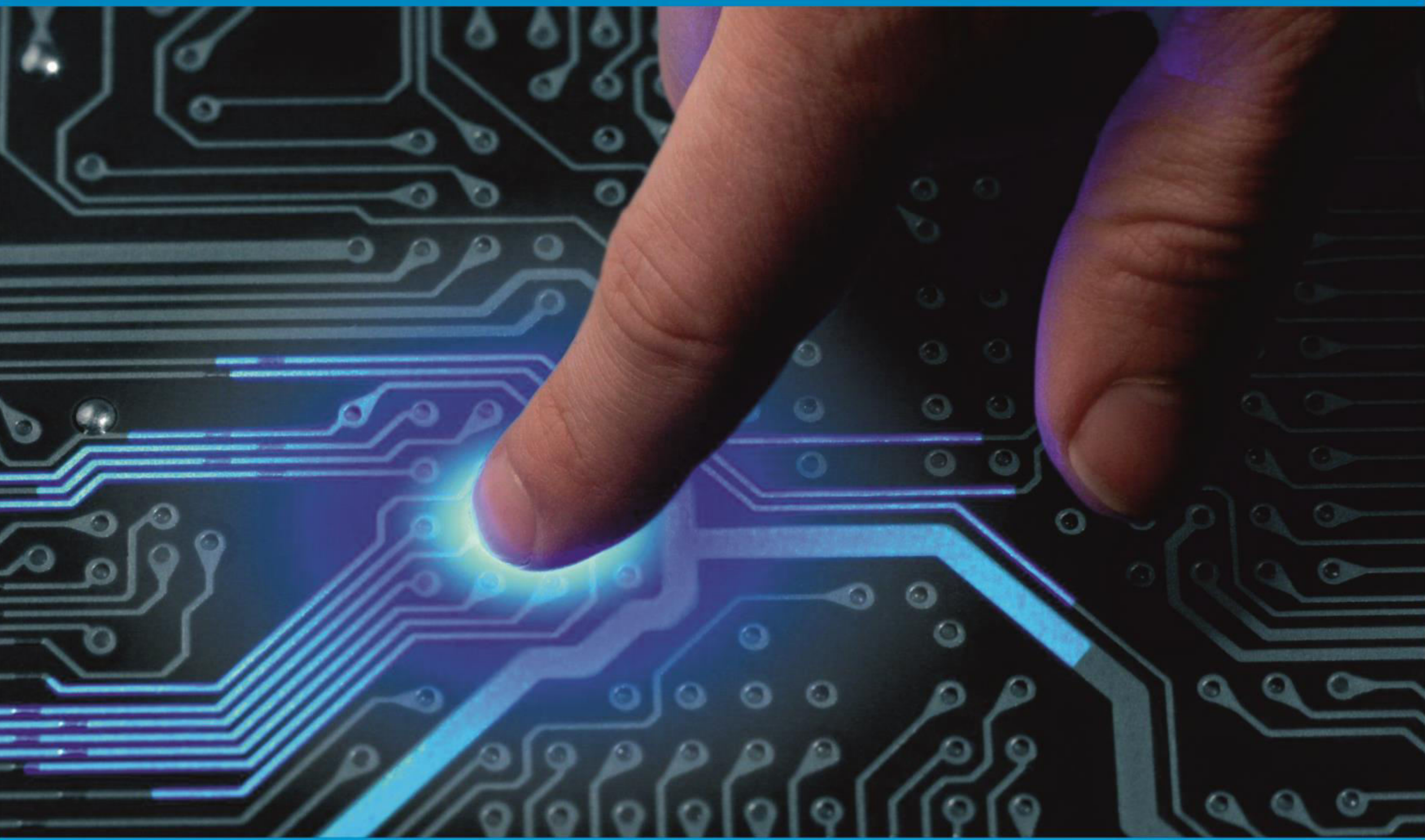
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 3, March 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.488

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903017 |

An LSTM Approach to Predict Rainfall

Hareeshma Jayakumar*[1], Anjana T.Bhaskar*[2], Harita Nair*[3], Gokul KG*[4], Suraj Rajappan[5]

Student, Dept. of Electronics and Communication Engineering Vidya Academy of Science and Technology

Thrissur, Kerala, India[1-4]

Assistant Professor, Dept. of Electronics and Communication Vidya Academy of Science and Technology

Thrissur, Kerala, India[5]

ABSTRACT: Rainfall Prediction is the application of science and technology to predict the amount of rainfall over a region. One of the important reasons to exactly determine the rainfall is for the effective use of water resources, crop productivity, and preplanning of water structures. But here the main focus is to relate the rainfall with one of the most destructive natural disasters i.e., flood. Flood prediction is one of the most challenging and uncertain tasks which has a significant impact on human society. Timely and accurate predictions can help to proactively reduce human and financial loss. To mimic the complex mathematical expressions of physical processes of floods, during the past two decades, machine learning (ML) methods contributed highly to the advancement of prediction systems providing better performance and cost-effective solutions. This study aims to predict daily rainfall with the help of a Long Short-Term Memory (LSTM), a kind of Recurrent Neural Network (RNN) considering the long term. The LSTM model was constructed and trained with the Keras neural network API [2]. With an LSTM model using the rainfall data from 1998 to 2018, a model is developed to predict the rainfall in Kerala which could be used for flood prediction in the area. Mean Squared Error(MSE), and Root Mean Squared Error(RMSE) were calculated after comparing the predicted values with the actual.

KEYWORDS: Rainfall, machine learning, LSTM, Keras, Kerala, MAE, MSE, RMSE

I. INTRODUCTION

Floods are the most common and widespread of all weatherrelated natural disasters. It can occur within minutes or over a long period and may last days, weeks, or longer. The primary effects of flooding include loss of life and damage to buildings and other structures, including bridges sewerage systemroadways, and canals. Damage to roads and transport infrastructure may make it difficult to mobilize aid to those affected or to supply emergency health treatment. Floodwaters typically inundate farmland, making the land unworkable and preventing crops from being planted or harvested, which may cause shortages of food both for humans and livestock . Entire harvests for a rustic are often lost in extreme flood circumstances. The research on the advancement of flood prediction models contributed to risk reduction, policy suggestion, minimization of the loss of human life, and reduction of the property damage related to floods. To mimic the complex mathematical expressions of physical processes of floods, during the past 20 years, machine learning (ML) methods contributed highly to the advancement of prediction systems providing better performance and cost-effective solutions. Due to the vast benefits and potential of ML, its popularity dramatically increased among hydrologists [1]. Robust and accurate prediction contributes highly to water resource management strategies, policy suggestions and analysis, and further evacuation modeling [3]. Aim Our main focus is to study the effectiveness of the LSTM algorithm for the prediction of rainfall. Hence the question is not whether they are effective or not, but how accurate an LSTM model can predict rain. The reason we choose LSTM is because of the abilities of the algorithm to consider long-term dependabilities thus making it a good algorithm for time-series predictions. Machine Learning Machine learning, also referred to as ML, is a tool for processing and finding patterns in data. Machine learning algorithms focus on optimizing and finding regularities in the data. A big part of machine learning consists of mathematics, the mathematics used is statistical learning and optimizationof the data. A big risk factor in machine learning is the risk of overfitting. Overfitting is when the algorithms rely too much on the sample and historical data, placing an all too high weight on the historic parameters, and eventually overfitting the data. The problem occurs when the prediction is being made of unknown data, the result is a high error rate due to the overfitting. Therefore most scientists and researchers work with hybrid methods on machine learning to minimize this risk as much as possible. Another important aspect related to machine learning that one needs to take into consideration is the representation of data. The data needs to be represented in a structured and accessible manner. The structure is

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903017 |

important because it can help the algorithm to process the data correctly [10]. Neural network (NN) is a technological concept capturing the scope of machine learning and the biological brain. The neural networks are built up by a network of neurons or nodes. The lines connecting the nodes are named edges. The nodes in the network receive information as input from the edges in the network. The information inputted into the nodes is multiplied by the weight set by the constructor of the network. The weight is used to adjust the importance of different computational results from the given node. The non-linear function is then added to the result, this function is referred to as the activation function. Typically the activation function is a tan(h) function and regulates how much of the weighted result from the neurons are added to the output, which can be the final output or an output passed on to more neurons or nodes in the network, in such a case our network will have multiple layers [11]. RNN is a class of neural networks that have the ability to have memory, which makes them more similar to how we humans process information and it is an efficient way to solve various scientific problems. In a traditional neural network, the data is processed independently. RNN is better than traditional neural networks in the case of predicting the next word in a sentence thanks to their ability to have a memory and recognize the context. LSTM is a kind of RNN, that has the ability to consider the long term dependability. LSTM was developed by two scientists, Schmidhuber and Hochreiter in 1997. What sets LSTM's apart from other RNN approaches is that LSTM's have the ability to remember information for a longer time period and avoid the long term dependabilities. LSTM's have a chain structure and on the inside, they operate using gates and layers of neural networks like other RNN approaches. The structure of the LSTM is constructed in a manner of a cell state that runs through the entire LSTM, the value is changed by the gates that have functioned by either allowing or disallowing data to be added to the cell state. There are also components by the name of gated cells that allow the information from previous LSTM outputs or layer outputs to be stored in them, this is where the memory aspect of LSTM's kick in [12].

II. METHODS

A model was implemented using the Keras framework and trained using a dataset that contained data about the daily rainfall level.

A. Data Collection and Exploratory Data Analysis

The source of the dataset is rainfall which can be measured either by ground rain gauges, or relatively new remote-sensing technologies such as satellites, multisensor systems, and/or radars. The dataset containing precipitation level of Thrissur for the past 20 years(1998-2018) was collected. Exploratory Data Analysis(EDA) refers to the critical process of performing initial investigations on data to discover patterns to spot anomalies to test hypotheses and to check assumptions with the help of summary statistics and graphical representations. It employs a variety of techniques (mostly graphical) to:

1. Maximize insight into a data set
2. Uncover underlying structure
3. Extract important variables
4. Detect outliers and anomalies
5. Test underlying assumptions
6. Develop parsimonious models
7. Determine optimal factor settings

Most EDA techniques are graphical in nature with a few quantitative techniques. The reason for the heavy reliance on graphics is that by its very nature the main role of EDA is to open-mindedly explore, and graphics gives the analysts unparalleled power to do so, enticing the data to reveal its structural secrets, and being always ready to gain some new, often unsuspected, insight into the data. The particular graphical technique included as EDA was plotting the raw data. Histograms and scatter plots were used to draw some useful insights. Figure 1 shows the raw data and figure 2 represents a sample after data scrubbing.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903017 |

Daily Rainfall data 1998-2019

date	1998	1999	2001	2002	2003	2005	2006	2007	2009	2010	2011	2013	2014	date	2000	2004	2008	2012
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7	0.0	0.0	0.0	0.0
8/1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8/1	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	13	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	14	0.0	0.0	0.0	0.0
15/1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15/1	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	16	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	18	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	19	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	20	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	21	0.0	0.0	0.0	0.0
22/1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	22/1	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	23	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	24	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	25	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	26	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	28	0.0	0.0	0.0	0.0
29/1	0.0	0.0	0.0	0.0	0.0	0.0	20.0	0.0	0.0	0.0	0.0	0.0	0.0	29/1	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0	0.0	7.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	30	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	31	0.0	0.0	0.0	0.0
1/2	0.0	0.0	12.2	0.0	23.6	0.0	6.0	0.0	0.0	0.0	0.0	0.0	0.0	1/2	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	3	0.0	0.0	0.0	0.0

RF data | RFdata

Fig.1 Daily rainfall data(in mm)

Date	Rainfall(mm)	Date	Rainfall(mm)
1/7/2000	35.0	1/7/2000	35.0
2/7/2000	21.4	2/7/2000	21.4
3/7/2000	17.0	3/7/2000	17.0
4/7/2000	30.8	4/7/2000	30.8
5/7/2000	20.5	5/7/2000	20.5
6/7/2000	0.6	6/7/2000	0.6
7/7/2000	5.2	7/7/2000	5.2
8/7/2000	9.5	8/7/2000	9.5
9/7/2000	21.8	9/7/2000	21.8
10/7/2000	7.8	10/7/2000	7.8
11/7/2000	1.6	11/7/2000	1.6
12/7/2000	37.4	12/7/2000	37.4
13/7/2000	108.8	13/7/2000	108.8
14/7/2000	17.2	14/7/2000	17.2
15/7/2000	0.0	15/7/2000	0.0

Fig. 2 Sample after data scrubbing

B. Data Preprocessing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Realworld data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Preprocessing is necessary before training the machine learning models. It removes outliers and scales the features to an equivalent range. Before applying the collected dataset to the LSTM model, normalization was done, with the insight for improved accuracy. We used Scikit- Learn'sMinMaxScaler and scaled our dataset to numbers between zero and one. The normalized dataset after splitting into test and train sets were plotted using the matplot library functions (shown in figure 3).

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |
|| Volume 9, Issue 3, March 2021 ||
| DOI: 10.15680/IJIRCCE.2021.0903017 |

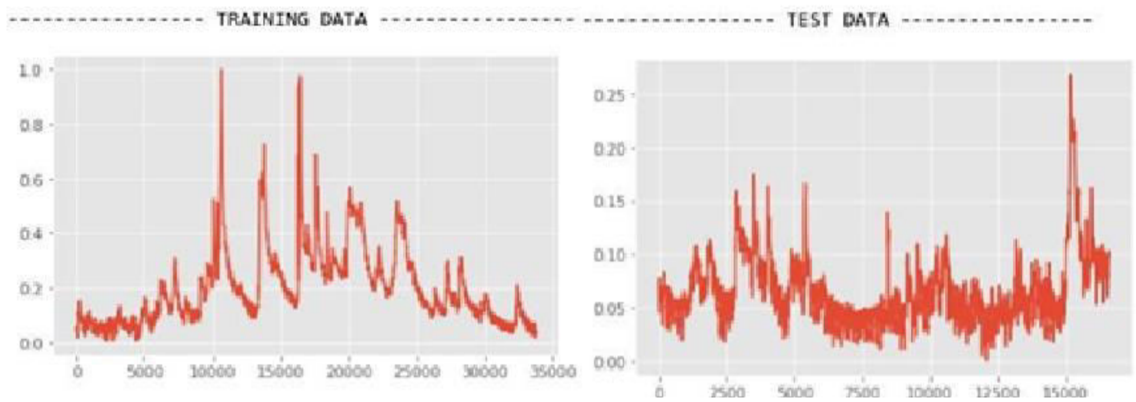


Fig.3 Normalized training and test data plotted using matplotlib

C. Frameworks

The LSTM model was constructed and trained with the Keras neural network API [2]. The API is an open-source deep learning library written in Python and uses Tensorflow as a backend. Tensorflow is also an open source machine learning framework for numerical computations [4]. Keras fast learning curve and together with its easy implementation of deep learning models made a great tool for us to use for this project.

III. PROPOSED MODEL

A. LSTMArchitechture:

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn. In figure 4, a chunk of neural network, A, looks at some input $x_t$and outputs a value h_t. A loop allows information to be passed from one step of the network to the next.LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn. In figure 4, a chunk of neural network, A, looks at some input $x_t$and outputs a value h_t. A loop allows information to be passed from one step of the network to the next.

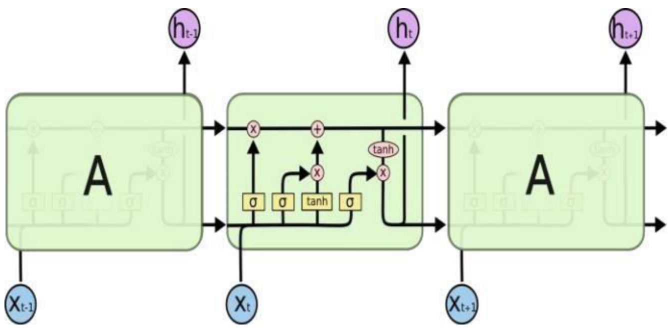


Fig. 4 LSTM Architecture

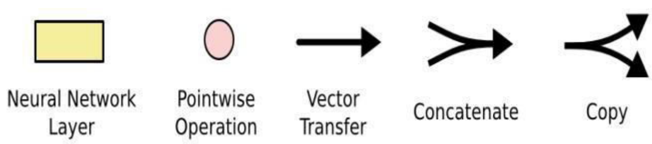


Fig. 5 Detailed representation

In figure 5, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903017 |

Lines merging denote concatenation, while a line forking denote its content being copied and the copies going to different locations.

B. LSTM Structure

The LSTM model's performance depends solely on the hyperparameter settings and the amount of training. The settings of the model are presented below.

Input

Together with all of the days and their feature which is included in the prediction we constructed a matrix and split all of the days into windows of a fixed size length. Then we reshape all the vectors in the matrix so that we get a NumPy array that is now a 3D vector of the shape (W ⇥ l ⇥ f), where W = Number Of Windows, l = Length Of A Window, and f = Number Of Features. This partitioning of the dataset was done to calculate the output easier and to support the input format of our framework Keras.We split the dataset into train and test sets where the training set contained 67% of the dataset (water level) and the rest (33%) were the test set. We then train the LSTM Model to determine the water level for each trainingday.

Layers

When constructing the LSTM model we have to take into consideration how many hidden layers the model will contain, the amount of LSTM cells that should be included in every layer, and what the dropout should be. The number depends on what application the LSTM model is going to be used on, so the number of cells and layers can differ but the layers are often from 1 to 5 and the cells in each layer should contain the same amount of cells for finding an optimalstructure. A dense layer is a densely connected NN layer [14], where a dense layer connects each cell to another in the next layer. We have seen successful models using dense layer by building their model of hidden layers followed by multiple dense layers.

Number Of Layers

We have decided that our LSTM model will contain four layers, i.e, three hidden layers and one dense layers. The output of the first hidden layer is connected to another hidden layer than that layer is connected to a dense layer, i.e.hiddenlayer ! hidden layer ! hidden layer ! dense layer, where ! represent the connection between layers. Dropouts are used after each hidden layer for preventing the risk of overfitting.

*Backtesting*Randomly selecting training, validation and testing sets in a data doesn't work for time series. It doesn't reflectthe sequential discovery process of time series and creates a huge risk. Prediction accuracy is the only true measure of performance for time series modeling, whether explicitly working on a prediction challenge, or on clustering, simulation, anomaly detection. The idea of backtesting is simple: at every moment in the data set, we have to train the model on known/past data at that moment, and test it on unknown/future data at that moment. Here backtesting is done using sliding window of length10.

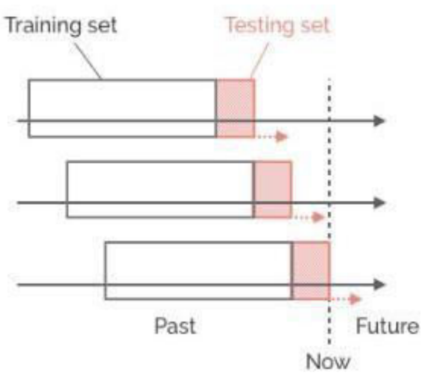


Fig. 6 Backtesting with sliding window

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903017 |

C. Optimal Hyperparameters

Number Of Epochs

An epoch is when the whole training data has been passed through the network, hence one epoch is one iteration of the whole training data being passed through the network. When the training data are being propagated through the network we split the training data into a batch size, where we defined the batch size as 10. This means that first the 10 samples are being taken from the training data (0-9) and trained on the network, then it takes the next 10 samples and trains the network. The epoch continues until all samples are propagated through the network hence then one epoch had been passed through the network.

Window

We then do an empirical test for finding the optimal length of a window, we do not want a short window size because then the model will not acquire the longer dependencies hence neglecting important information. Thereis also a downside with a window size that is too large, it will then add a larger amount of redundant noise hence will overfit the training data. The best way to find the most suitable window size is to do an empirical test on different window size and find what value has the smallest MSE on the training data while training the LSTM model. We can see that the optimal length for the window in our case is10.For measuring the accuracy of prediction based on our data and LSTM model, we have set aside like described before 33% of the rainfall data as the test data. We have normalized the water level and that will be compared to the LSTM Models prediction when backtesting the test data. When backtesting the data we can see if the model that is built is overfitting [16], where this problem happens when the model is memorizing the data in our case, water level instead of learning the patterns.

Optimizer

When building the LSTM model we used the optimizer Adam because of its high performance and fast convergence compared to other alternative optimizers [5] and it was recommended to use it as default. When using the optimizer Adam we set the decay to 0.3.

Error Calculation

We then will evaluate the MSE and RMSE when evaluating the LSTM model's train score and test score, where the score is the evaluation of our chosen loss function which in our case is MSE. The MAE will also be calculated during the backtesting of the data for ensuring that the model is predicting with highaccuracy.

Mean Square Error (MSE) is a method of computing the accuracy and the error in the predictive models used.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - y_i)^2$$

where y_i is the ith predicted value and Y_i is the ith actual/observed value.

Mean Absolute Error (MAE) of a model refers to the mean of the absolute values of each prediction error on all instances of the test data-set.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Ai - Fi}{Ai}\right|$$

where Ai is the actual value and Fi is the forecast/predicted value.

Root Mean Square Error (RMSE) is a method to calculate the error or accuracy in the prediction of ones models. The RMSE calculates the error based on standard deviations. The final output is given in a standard deviation of the magnitude of the error, the individual calculations are outputted as residuals. [17].

$$RMSE = \sqrt{MSE}$$

IV. PSEUDO CODE

STEP 1 : Import the libraries.
STEP 2 : Fix the random seed for reproducibility.
STEP 3 : Read the data (csv file).

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903017 |

STEP 4 : Plot the dataset using matplotlib.
STEP 5 : Normalize the data set.
STEP 6 : Split the dataset in to train and test sets. The ratio should be 70:30 or 75:25.
STEP 7 : Plot the training data set and test set separately.
STEP 8 : Create a sliding window of the dataset.
STEP 9 : Setup the LSTM model.
STEP 10: Print out the evaluation score for both train and test set.
STEP 11: Shift and plot the predictions.
STEP 12: Test the network on an unseen dataset.
STEP 13: Check the root mean squared error for the new test set.

V. RESULTS

For measuring how accurate our LSTM model is, we will make a prediction based on past data to make a prediction of the future history data. This will be done by feeding the LSTM model the last prediction that was feed to the LSTM model, which will be the last entry data of the training set, wewill then make a prediction which will represent the next water level. We then save that predicted water level which we can call P0 and starting to predict the next water level by feeding P0 to the LSTM model we will get the predicted water level P1. This iterative process will continue stepwise until we have predicted the number of days data as we want. When we then have predicted the water level over a certain amount of days, we will plot the prediction for days and the real water level during that time. In this case, we can see if our model has overfitted the data during the backtesting because if the plot of the prediction is not near the backtesting then we will know that our model has overfitted. Training was done using the rainfall data from 1998 to 2003 and predicted daily rain for four years (2003 to 2006). We compared the obtained result with the actual values. Fig 7 shows the plot of actual rainfall values and in fig 8, a plot of predicted values for daily rain is shown.Table 1 shows the average RMSE, MAE, and MSE values obtained after evaluation.

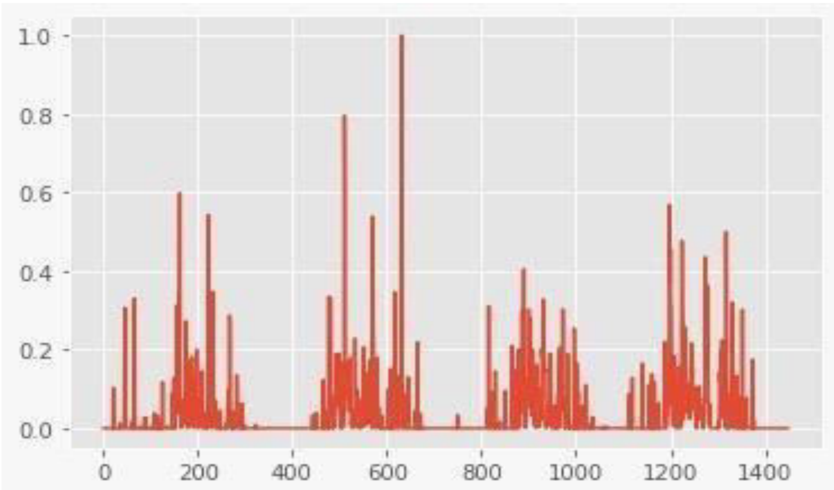


Fig. 7 Actual rainfall

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903017 |

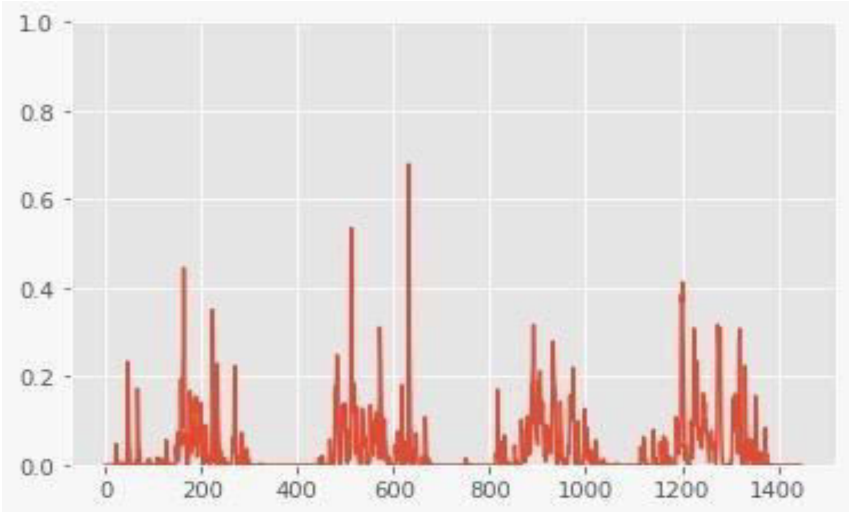


Fig. 8 Plot of predicted daily rain

Table 1

Performance Parameter	Values
RMSE	0.076344
MAE	0.034806
MSE	0.005828

Table 1 Average RMSE, MAE and MSE

VI. CONCLUSION AND FUTURE WORK

We believe our conclusion is rather important and can assist other scientific works in this field, where models are built and accuracy is measured for predictions. If one reads some paperswritten by other scientists in this field, one will be amazed by the overall high accuracy in all the models. Our conclusion leads us to question these results if they solely contain back tested data. Thus our recommendation is to always do a prediction based on real-time data and measure that in complement with the back tested data results, because of the risk of overfitting. Even when deploying certain risk- management techniques to combat overfitting such as optimizing a hyperparameter we realized that overfitting is still present as a risk in the accuracy of themodel.

In this field, there are several components that we believe other researchers should adopt to their frameworks of scientific work. The first part would be optimizing a hyperparameter to deal with the problem of overfitting when back testing the data. The second aspect would be to include more features like the land pattern and weather conditions which also is a reason for severe climatic changes that leads to natural disasters like flood. The third aspect would be to create a hybrid model where we can arrive at more useful insights and thereby leading to improvisations in agriculture or related field.

REFERENCES

1. Flood Prediction Using Machine Learning Models: LiteratureReviewAmirMosavi1 , Pinar Ozturk 1,and Kwok-wing Chau 2.
2. Keras. Keras: The python deep learning library, 2018a. URL https://keras.io[Accessed: 2018-03-12].
3. Xie, K.; Ozbay, K.; Zhu, Y.; Yang, H. Evacuation zone modeling under climate change: A data-driven method.J. Infrastruct. Syst. 2017, 23, 04017013.
4. Tensorflow. An open-source machine learning framework for everyone, 2018. URL https: //www.tensorflow.org[Accessed:2018-03-12].

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903017 |

5. Nils Reimers and Iryna Gurevych. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. arXiv preprint arXiv:1707.06799, 2017.
6. Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. arXiv preprint arXiv:1409.2329, 2014.
7. Aziz, K.; Rahman, A.; Fang, G.; Shrestha, S. Application of artificial neural networks in regional flood frequency analysis: A case study for australia. Stoch. Environ. Res. Risk Assess. 2014, 28, 541–554.
8. Campolo, M.; Andreussi, P.; Soldati, A. River flood forecasting witha neural network model. Water Resour. Res. 1999, 35,1191–1197
9. Mackey,B.P.;Krishnamurti,T.Ensembleforecastofatyphoonfloodevent.Weat her Forecast. 2001, 16,399–415.
10. Nasser M Nasrabadi. Pattern recognition and machine learning. Journal of electronic imaging, 16(4):049901,2007.
11. Jürgen Schmidhuber. Deep learning in neural networks: An overview. Neural networks, 61: 85–117,2015.
12. Sepp Hochreiter and Jürgen Schmidhuber. Long short-termmemory. Neural computation,9 (8):1735–1780,1997.
13. Alex Graves and JÃrgenSchmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks, 18(5):602–610, 2005. ISSN0893-6080.
14. Keras. Dense, 2018b. URLhttps://keras.io[Accessed:2018-03-24].
15. W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont,CA:
16. Wadsworth, 1993, pp. 123–135.
17. David H Bailey, Jonathan M Borwein, Marcos Lopez de Prado, andQiji Jim Zhu. The probability of backtest overfitting.2016b.
18. Cort J Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. Climate research, 30 (1):79–82,2005..
19. J. Wang, “Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication),” *IEEE J. Quantum Electron.*, submitted forpublication.
20. Daud, A., Zakaria, R.A.A., Sahat, S., Ismail, N.F.A., Mohamad, N.F., Rosli, N.: The Study of Thunderstorm and Rainfall Occurrences over Pahang (in the Period 1998–2012). Malaysian Meteorological Department, Petaling Jaya(2015)
21. Shoaib, M., Shamseldin, A.Y., Melville, B.W., Khan, M.M.: A comparison between wavelet based static and dynamic neural network approaches for runoff prediction. J. Hydrol. 535, 211–225(2016)
22. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO,private communication, May1995.
23. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces (Translation Journals style),” *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740–741 [*Dig. 9thAnnu. Conf. Magnetics* Japan, 1982, p.301].
24. M. Young, *The Techincal Writers Handbook*. Mill Valley,CA: University Science,1989.
25. J. U. Duncombe, “Infrared navigation—Part I: An assessment of feasibility (Periodical style),” *IEEE Trans. Electron Devices*, vol. ED- 11, pp. 34–39, Jan.1959.
26. S. Chen, B. Mulgrew, and P. M. Grant, “A clustering technique for digital communications channel equalization using radial basis function networks,” *IEEE Trans. Neural Networks*, vol. 4, pp. 570–578, Jul.1993.
27. Mekanik, F., Imteaz, M.: A multivariate artificial neural network Approach for rainfall forecasting: case study of Victoria, Australia. In: Proceedings of the World Congress on Engineering and Computer Science(2012)