Flood Prediction using Deep Learning Models

Muhammad Hafizi Mohd Ali, Siti Azirah Asmai*, Z. Zainal Abidin, Zuraida Abal Abas, Nurul A. Emran
Centre for Advanced Computing Technology (C-ACT), Fakulti Teknologi Maklumat Dan Komunikasi
Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

***Abstract*—Deep learning has recently appeared as one of the best reliable approaches for forecasting time series. Even though there are numerous data-driven models for flood prediction, most studies focus on prediction using a single flood variable. The creation of various data-driven models may require unfeasible computing resources when estimating multiple flood variables. Furthermore, the trends of several flood variables can only be revealed by analysing long-term historical observations, which conventional data-driven models do not adequately support. This study proposed a time series model with layer normalization and Leaky ReLU activation function in multivariable long-term short memory (LSTM), bidirectional long-term short memory (BI-LSTM) and deep recurrent neural network (DRNN). The proposed models were trained and evaluated by using the sensory historical data of river water level and rainfall in the east coast state of Malaysia. It were then, compared to the other six deep learning models. In terms of prediction accuracy, the experimental results also demonstrated that the deep recurrent neural network model with layer normalization and Leaky ReLU activation function performed better than other models.**

Keywords*—Deep learning; recurrent neural network; long short-term memory; flood prediction; layer normalization

I. Introduction

Due to its impact on daily life, flooding is one of the most pressing issues that Malaysia has been dealing with recently. Floods are a type of natural geohazard that typically occur because of consistently heavy rain. This natural phenomenon causes massive damage to the country's property and Gross Domestic Product (GDP). According to Ashizawa et al. [1], the entire GDP of Japan impacted by flood damage is at least 1% of the overall GDP of the nation. Tiggeloven et al. [2] stated that the top 15 countries, such as India, Bangladesh, China, and others, are vulnerable to flood occurrence at the present day and could be worst if no action is taken. Indeed, floods can cause a massive amount of money to repair the damage. Hence, flood occurrence can affect every country, including Malaysia. Shaari et al., [3] stated that from 2006 to 2010, there was nearly 1 million USD damage caused by floods which affected the nation's economic growth. There are many classifications of floods namely coastal floods, flash floods, ponding (or pluvial flooding), and river (or fluvial) floods [4]. Floods often occur, especially in Southeast Asia, including our country, Malaysia. The general types of flooding in Malaysia include riverbank overflow, flash floods, high tides [5], and monsoon floods [6].

Floods are classified as natural disasters in Malaysia due to the monsoon season. In Malaysia, there are two distinct monsoon seasons: the Northeast Monsoon, which occurs from November to March, and the Southwest Monsoon, which occurs from late May to September. The Northeast Monsoon can bring heavy rainfall. Due to the extensive network of rivers connecting several Malaysian states and the country's poor drainage, floods are expected in Malaysia during the monsoon season, especially on the West Coast and in Borneo. The populous region is flooded as a result of the river level rising significantly as a consequence of these strong rains. As a result, people are compelled to temporarily relocate to several relief. Floods halt economic progress since crops and animals are destroyed. Romali et al, [7] stated that Malaysian financial losses are estimated at nearly MYR 915 million annually on an average due to floods.

According to Zerara [8], time series is a statistical method that can be applied in a broad range of longitudinal research designs. Typically, this time series design involves a single subject that is measured repeatedly at regular intervals over a large number of observations. Time series forecasting aims to predict an outcome based on the collection of historical data that can be used to build a quantitative model that explains the variables under consideration [9]. For many years, time series forecasting has been an important research domain in meteorology [10], biology [11], and econometrics [12]. Generally, time series can have four characteristics: trends, seasonality, cycles, and noise [13]. Time series forecasting algorithms perform well with data that includes a time dimension and one or more properties [14].

Time series have been frequently utilized in flood forecasting and have shown excellent results for the global community [15]. Furthermore, according to Shen et al. [16], modern time series can be combined with deep learning models including Recurrent Neural Network (RNN), Artificial Neural Network (ANN), Long Short Term Memory (LSTM), and other models. However, the existing flood forecasting methods, for example, frequency analysis, rational method, and empirical formula, are not deemed suitable for a wide area. Those methods can only cover a small river flow area [17]. For example, Faruq et al. [18] used an LSTM model to predict the flood by using a Klang River lever dataset. Another study [19] used the ANN model to predict floods by using the Kelantan river lever and rainfall dataset in separate models. The LSTM model performs exceptionally well when the modelling has a large amount of time series data. Furthermore, according to the literature, the LSTM model outperforms the RNN in the number of previous time steps that can be considered. Additionally, Siami-Namini & Namin, [20] demonstrated that the LSTM model could predict time series much more accurate than the Autoregressive Integrated Moving Average (ARIMA) model in some cases. As a result, the LSTM model appears to be a likely top performer; although the study employs stock time series, it still has parallels to the current study since it focuses on time series forecasting. According to Jaiswal & Das

*Corresponding Author.

[21], the ANN model works best with nonlinear problems whereas Šiljić Tomić et [22] stated that the ANN model can work with both nonlinear and linear problems. According to Y.f. Zhang et al. [23], despite being the most significant advantage of time series, long-term dependencies remain a considerable challenge. Besides, one of the most common drawbacks of the LSTM model and BI-LSTM model is the high computational cost during training procedures [24], where potential time series forecasting in floods has yet to be unfolded. Despite advances in developing models based on RNN, these models remain challenging to scale to long data sequences. Dhunny et al. [25] have proven that an ANN model can predict the flood water level well within 24 hours ahead of time by using the data from rainfall and present river level data. In this study, the flood was predicted for one day ahead.

The rest of this paper is structured as follows. Section II describes the related work and literature for this study. Section III presents the proposed models for flood prediction. Section IV covers the experimental procedure for the case study whereas the findings are discussed in Section V. Finally, Section VI concludes the paper, and a discussion of future works is included in Section VII.

II. Related Work

Artificial neural networks, often known as deep learning, are machine learning algorithms that have been influenced by the structure and operation of the human brain. Deep learning has dominated many uses and has proven to be superior to traditional machine learning algorithms because it can produce faster and more accurate results [26].

Deep learning enables computational models comprising multiple hidden layers of artificial neural networks with multiple linear and nonlinear transformations that learn the data representations with multiple abstraction levels [27]. In deep learning, multiple layers of nonlinear processing units perform feature extraction transformation in the deep learning model. Every layer in the previous layer input is used as its output and it is used in both supervised and unsupervised methods for classification problems and pattern analysis problems [28]. The characteristic of neural networks can be seen in Fig. 1.

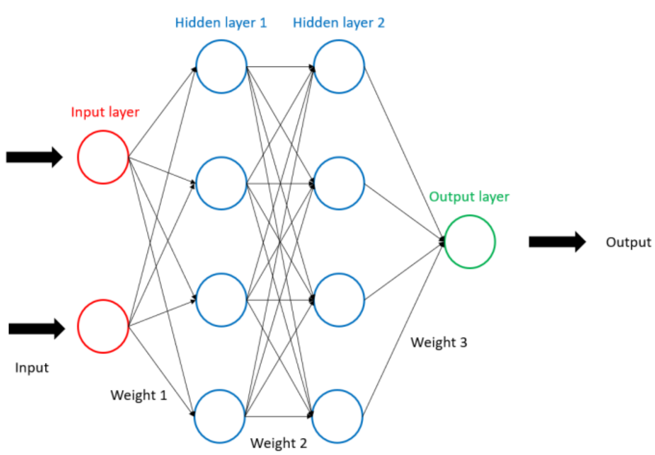

Fig. 1. A Characteristic of Neural Network.

The human brain structure inspires the neural network architecture. Our brains can be trained to recognise patterns and classify various types of information. The possibility of detecting and displaying the correct answer increases with each layer of a neural network, which may be thought of as a kind of filter that operates from coarse to fine.

The neural network can begin to identify trends across the many samples it processes and classify data based on their similarities by using several layers of functions to decompose unstructured data into data points and information that a computer can use.

After processing a large number of structured data training samples, the algorithm has created a model of which elements in data and their relationships must be taken into account when determining whether structured data is present or not. The neural network compares new data points to its model based on all previous evaluations when evaluating new data. The model is then used to determine whether the data contains specific data.

The layers of functions that are present between the input and output in this example serve as a representation of deep learning. The interaction across layers is marginally enhanced in the following Fig. 2, however, the connections between nodes or artificial neurons might vary significantly.

A. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks can be considered as recurrent neural networks that are modified to improve from the RNN model with memory recall function. The LSTM classifier is ideal for processing, classifying, and forecasting time series with unknown time lags. Backpropagation is applied when training the model. There are three gates in the LSTM model as shown in Fig. 3.

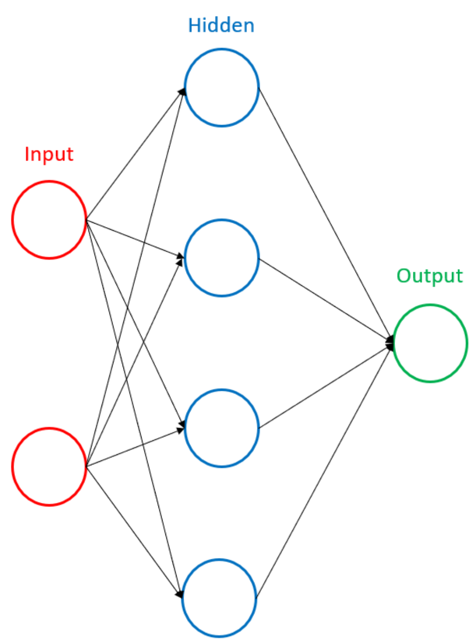

Fig. 2. Interaction between Layers.

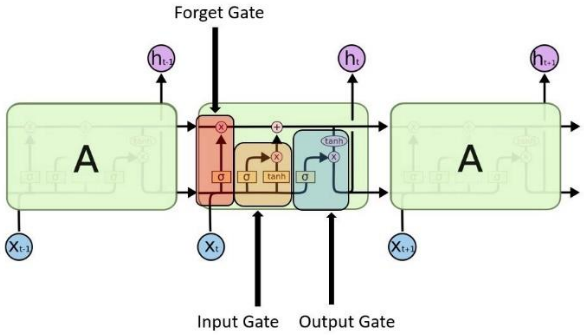

Fig. 3. LSTM Gates [29].

Forget gate - it analyses the previous state *(ht-1)* and the input content *(Xt)* and returns the value ranging between 0 to 1 for each number in the cell state *Ct-1* by deciding through the sigmoid function. The forget gate equation of [30] can be referred to as (1).

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{1}$$

Input gate - values from the input are applied to adjust the memory. The sigmoid function determines the value between 0 and 1, allowing through. The tanh function gives weightage to the values and then passes it, specifying their significant level ranging between -1 and 1. The input gate equation of [30] can be referred to as (2).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$Ct = ft \times Ct - 1 + it \times C't \tag{2}$$

Output gate – to determine the output, the memory and input of the block are applied. The sigmoid function determines the value between 0 and 1, allowing through. The tanh function gives weightage to the values and then passes it, specifying their significant level ranging between -1 to 1 and multiplying it by the sigmoid output. The forget gate equation of [30] can be referred to as (3).

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t) \tag{3}$$

B. Recurrent Neural Network

A recurrent neural network (RNN) is a simplified feedforward neural network with internal memory as seen in Fig. 4. RNN is recurrent because it works with the same function for each input data, and the current input as output is dependent on the previous computation. The output is produced, replicated, and then returned to the recurrent network. When making a decision, the current input and output might be seen as learned from the past.

By taking advantage of the internal state (memory) that the RNN model has, it can process the input sequence that is different from typical feedforward neural networks. As a result, the RNN model is suitable for speech recognition, unsegmented, or handwriting recognition tasks. However, there is one drawback in the RNN model: a vanishing gradient problem when dealing with long sequence data. The inputs are completely independent of one another for other neural networks. On the other hand, all the inputs are connected in RNN.

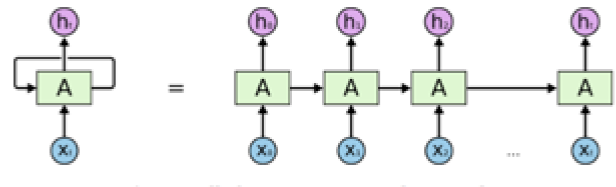

Fig. 4. An Unrolled RNN [29].

The *X (0)* is taken from the input sequence at first, then the output h (0), with *X (1)*, as the input for the next step. The *h (0)* and *X (1)* are the input for the next step. Similarly, *h (1)* from the previous step becomes the input for the next step of *X (2)*.

The formula equation of [31] can be referred to as (4) and this is the current state of the equation.

$$h_t = f(h_{t-1}, x_t) \tag{4}$$

Activation Function is applied:

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}X_t) \tag{5}$$

The equation of [31] can be referred to as (5), in which W is the weight, *h* is the single hidden vector, *Whh* is the weight at the previous hidden state, *Whx* is the weight at the current input state, and tanh is the activation function that applies a non-linearity squash of the activations to the range between [-1.1].

Output:

$$y_t = W_{\mathrm{hy}}\, h_t \tag{6}$$

The equation of [31] can be referred to as (6), in which the output state is *Yt*. The weight in the output state is *Why*.

Flood forecasting is a tool that allows flood control management to predict when local flooding is likely to occur with a high degree of accuracy. The river basin or watershed size can indicate the water levels and flow rates for intervals ranging from a few hours to days ahead; forecasted streamflow and precipitation data are used in the streamflow routing model and rainfall runoff. Flood forecasting can also use precipitation forecasts to expand the available lead time.

Flood forecasting is a crucial component of a flood warning system. The difference between flood forecasting and a flood warning is that flood forecasting produces a set of forecast time that profiles the river levels or flows channel at different locations. In contrast, "flood warning" refers to using forecasts to inform about flood warnings. A popular method applied for flood forecasting is hydrological modelling because this model is a simplified representation of a real-world system [32]. Although this model is good, the downside is that it is a scaling problem that faces a scale area parameter [33].

The existing flood forecasting method cannot be used with a traditional database based on a single source as the main data [34] and it requires a lot of data. With the current technology, flood prediction is more robust, and real-time flood forecasting in the provincial area can be accomplished quickly by utilising the technology of artificial intelligence (AI) and fourth industrial technology (4IR). An effective real-time flood forecasting model may be helpful for disaster prevention, offering an advanced alert and mitigating the damage from the flood occurrence [35]. Flood forecasting has been improved by utilising deep learning models such as LSTM, RNN, and many others [18]. Many studies have applied a deep learning model in their study to predict flood occurrence and are proven to be an informative and accurate model as shown in Table I. Hence, for the deep learning model, there is always room for improvement with the use of uncertain data like flood data with a nonlinear characteristic. However, there are very few studies that build a deep learning model for flood prediction using multivariate data. In addition, the majority of past research has only used one variable or input as their main source of information to forecast the flood. In addition, the previous research already demonstrates a high level of accuracy. However, the majority of these studies utilize basic deep learning models without any additional characteristics, which can be seen as a discrepancy between the studies. The proposed models aim to get a dependable and more accurate prediction model while reducing the limitations of the previous study by using an additional characteristic as described in Section III. As previous studies are concerned with using a single data as primary data, this study proposed multivariate data as primary data to determine a correlation between several variables simultaneously and a deeper understanding of how the multivariate data relate to real-world scenarios like flood occurrence.

TABLE I. LITERATURE SUMMARY

Models	**Title (Author and Year)**	**Goal**	**Country**
(LSTM) and Radial basis function neural network (RBFNN)	Deep Learning-Based Forecast and Warning of Floods in Klang River, Malaysia [18]	Forecasting the river water level in the Klang River basin, Malaysia.	Malaysia
LSTM and RNN	Application of Long Short-Term Memory (LSTM) neural network for flood forecasting [36]	Proposing an effective approach to flood forecasting based on the data-driven method.	Vietnam
ANN	Flood Prediction through Artificial Neural Networks: A case study in Goslar [37]	Establishing, training and evaluating a neural network for the detection of flood hazards and concrete water levels.	Germany
LSTM	Flood Prediction and Uncertainty Estimation Using Deep Learning [38]	Exploring the deep learning model for predicting gauge height and evaluating the associated uncertainty	United States of America
LSTM	Flash flood forecasting based on long short-term memory networks [39]	Forecasting a model based on (LSTM) for flash flood forecasting.	China
ARIMA and LSTM	Forecasting Economic And Financial TIme Series: ARIMA Vs LSTM [20]	Investigating which forecasting methods offer the best predictions with the lower forecast errors and higher accuracy of forecasts	Not stated
LSTM and TCN	An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modelling [40]	Raising issues of whether these successes of convolutional sequence modelling are confined to specific application domains or whether a broader reconsideration of the association between sequence processing and recurrent networks are in order.	Not stated
LSTM and TCN	Temporal Convolutional Networks Applied to Energy-related Time Series Forecasting [41]	Proposing a TCN-based deep learning model to improve the predictive performance in energy demand forecasting	Spain

III. PROPOSED MODELS

The proposed enhanced models for flood prediction were formulated to increase the prediction accuracy. In this study two methods in the models were introduced as follows:

A. Layer Normalization

Inspired by the results of Batch Normalization, the Layer Normalization method is proposed by normalizing activations along the feature direction rather than the mini-batch direction. Hence, overcoming the disadvantages of batch normalization by eliminating the reliance on batches and making it easier to apply for RNN. Each activation feature is normalized to zero mean and unit variance through Layer Normalization.

In Batch Normalization, the statistics are computed across the batch, as for the spatial dimensions. In contrast, Layer Normalization (LN) computes statistics (mean and variance) across all channels and spatial dimensions. As a result, the statistics are batch independent. This layer was initially designed to handle vectors (mainly the RNN outputs).

Layer Normalization visually comprehends this as shown in Fig. 5:

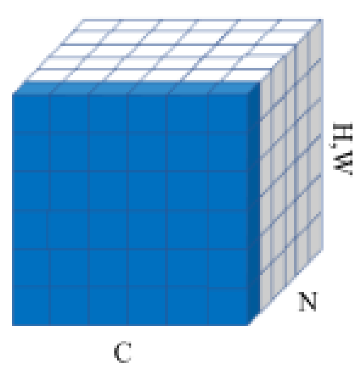

Fig. 5. An Illustration of Layer Normalization [42].

When dealing with vectors with a batch size of NN, the 2D tensors of shape R N times K RNK.

Normalize with the mean and variance of each vector because it does not depend on the batch and its statistics. The normalize equation of [43] can be referred to as (7).

$$\mu_n = \frac{1}{K}\sum_{k=1}^{K} x_{nk}$$
$$\sigma_n^2 = \frac{1}{K}$$
$$\sum_{k=1}^{K} (x_{nk} - \mu_n)^2 \tag{7}$$
$$\hat{x}nk = \frac{x_{nk} - \mu n}{\sqrt{\sigma_n^2 + \epsilon}}, \hat{x}nk \in R$$
$$\mathrm{LN}_{\gamma,\beta}(x_n) = \gamma \hat{x}_n + \beta, x_n \in R^K$$

When generalizing to 4D feature map tensors, it takes the mean across all channels and spatial dimensions, as shown below: The equation below based on [43] can be referred to as (8).

$$LN(x) = \gamma\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \beta$$
$$\mu_n(x) = \frac{1}{CHW}\sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W} x_{nchw} \tag{8}$$
$$\sigma_n(x) = \sqrt{\frac{1}{CHW}\sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W} (x_{nchw} - \mu_n(x))^2}$$

B. Leaky ReLU

To replace its saturated counterpart of Sigmoid or Tanh, the modern deep learning system employs a non-saturated activation function such as ReLU and Leaky ReLU. It solves the "exploding/vanishing gradient" issue and speeds up convergence.

ReLU reduces the negative component to zero while keeping the positive component. It has the desirable property of being sparse in activations after passing through ReLU. The equation of [44] can be referred to as (9).

$$y_i = \begin{cases} x_i & \text{if } x_i \geq 0 \\ 0 & \text{if } x_i < 0 \end{cases} \tag{9}$$

The gradient-based optimization algorithm will not change the weights of a unit that does not initially activate. Because the gradient is 0 when the unit is inactive, ReLU has a disadvantage during optimization.

If the neurons are not activated at the start of the ReLU, it is possible to end up with a neural network that never learns. The learning rate is slow when training ReLU networks with constant 0 gradients. The equation of [44] can be referred to as (10).

$$y_i = \begin{cases} x_i & \text{if } x_i \geq 0 \\ \frac{x_i}{a_i} & \text{if } x_i < 0 \end{cases} \tag{10}$$

Leaky ReLU adds a slight negative slope to the ReLU to sustain and keep the weight updates alive throughout the propagation process.

The alpha parameter was introduced to address the ReLUs dead neuron issues, ensuring that gradients are never zero during training.

The ReLU function and the Leaky ReLU function are nearly identical as seen in Fig. 6. During optimization, the Leaky ReLU foregoes hard-zero sparsity in exchange for a potentially more robust gradient. Alpha is a constant value (float >= 0.).

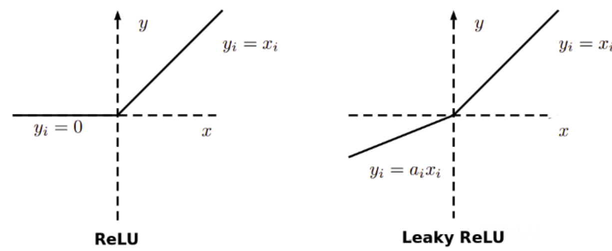

Fig. 6. ReLU vs Leaky ReLU [45].

Unlike the ReLU function, the Leaky ReLU has a non-zero gradient across its entire domain. The Leaky ReLU activation function is only available in the form of layers, not activations.

IV. Experimental Procedure

As shown in Fig. 7, the experiment design started with data collection and ended with a model evaluation.

A. Data Collection

A dataset of river level and rainfall at Rantau Panjang, Pasir Mas in Kelantan from 2013 until 2017 was used, as shown in Fig. 8 and Fig. 9. The data of these rivers were recorded every year with flood occurrence [46]. The data were provided by the Department of Irrigation and Drainage (DID) Malaysia, and the features variable is shown in Table II. The river at Pasir Mas station collected river level (m) and rainfall (mm) daily. This dataset contained one measurement per day. The dependent variable was observed as a single daily value; hence, these cloud values were summed, accomplished by taking the average between 00:00 and 24:00 as each day's value. Because there were fewer and more irregular observations, the cloud base was summarised by averaging an entire day.

The dataset was imported into pandas by using the read csv() function and saved in the Data Frame named "df.". Because the dataset was in tabular form, it was automatically converted into a Data Frame when working with tabular data in Pandas. In Python, a Data Frame is a two-dimensional, mutable data structure. It is made up of rows and columns, much like an excel sheet.

B. Data Cleaning

Data cleaning or filtering's main function was to correct (or remove) and detect inaccurate data in the dataset. The task involves identifying inaccurate, incomplete, incorrect or irrelevant parts of the data and then deleting the noise data, and replacing or modifying them [48].

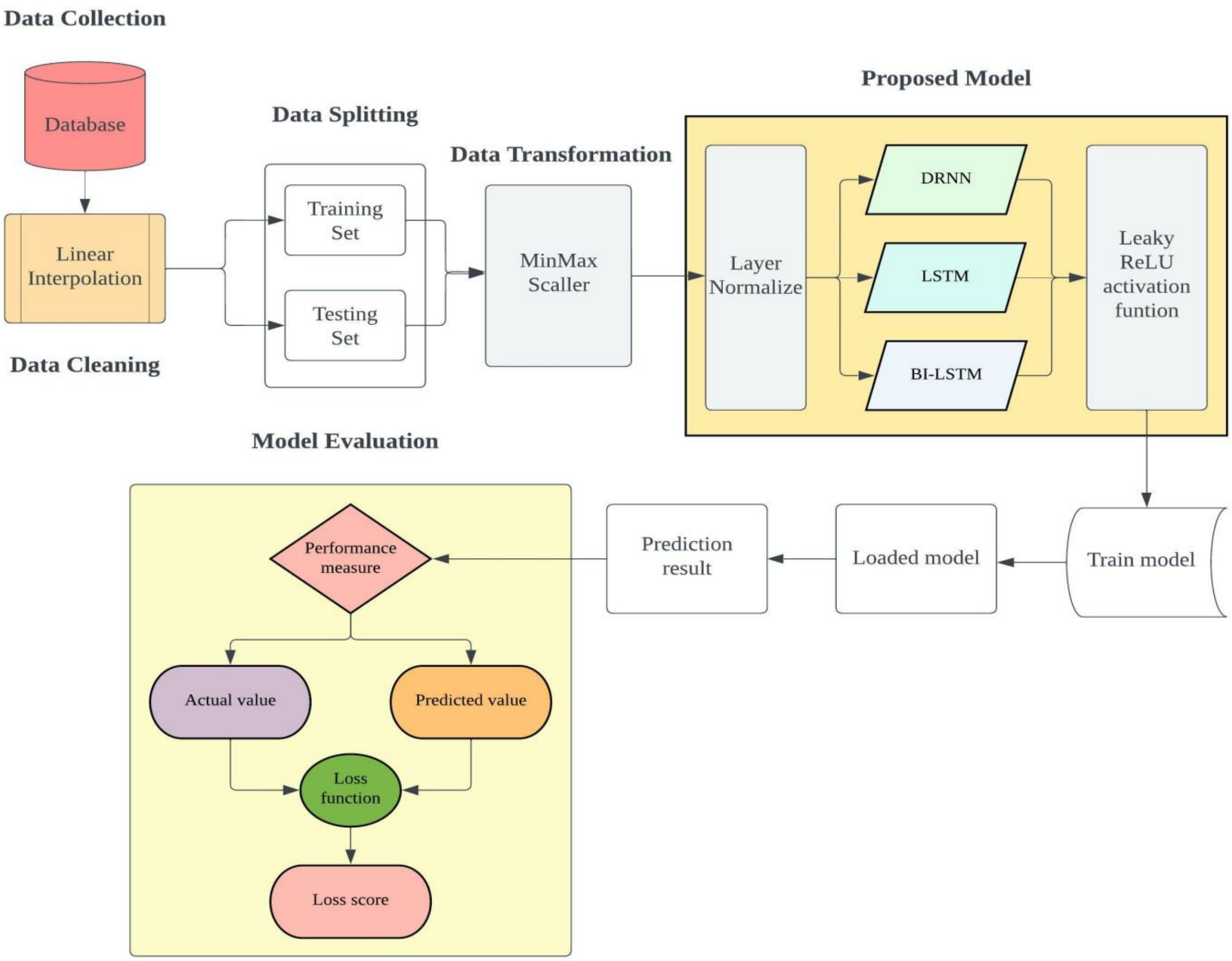

Fig. 7. Experiment Design.

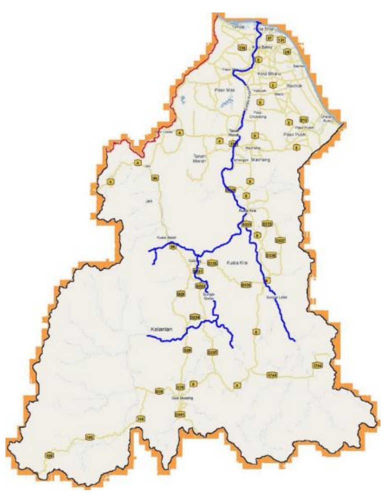

Fig. 8. Kelantan River Map [47].

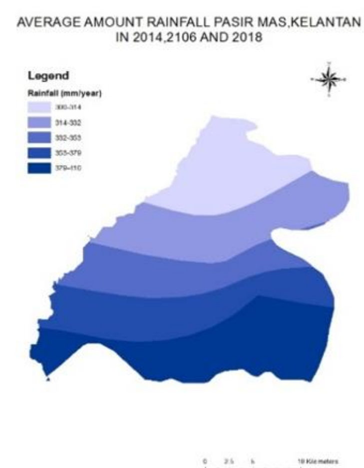

Fig. 9. Average Rainfall in Pasir Mas [46].

TABLE II. SUMMARY OF VARIABLE DATA COLLECTION

Variable	Unit	Description
Date	Date/ dd/mm/yyyy	Describe the current date
River Stage	Meter/m	Describe the river level during the day
Rainfall	Millimetres/mm	Describe the rain during the day

Data cleaning is the most crucial task because having incorrect or poor-quality data can harm processes and analysis. Clean data boosts overall productivity and allows for the utilization of the highest quality data when making predictions.

First, the river of Pasir Mas dataset needed to be checked to determine whether the dataset had missing or null values. After reviewing the dataset, a missing or null value in this situation needed to be dealt with through two options: a) removed a row with a missing value; b) replaced the missing value by using one of the most fundamental methods namely linear interpolation method. The formula is as follows:

$$y = y_a + (y_b - y_a)\frac{x-x_a}{x_b-x_a} \text{ at the point } (x, y)$$
$$\frac{y-y_a}{y_b-y_a} = \frac{x-x_a}{x_b-x_a} \tag{11}$$
$$\frac{y-y_a}{x-x_a} = \frac{y_b-y_a}{x_b-x_a}$$

The equation of [49] can be referred to as (11), the new line slope between (x_a, y_a) and (x,y) is the same as the slope of the line between (x_a,y_b) and (x_b,y_b). The advantage of linear interpolation is that it is easy and fast to be applied, but its accuracy is doubtful.

The river of Rantau Panjang Pasir Mas station had ten missing values in the river level variable because of the existence of the same reading of recording on certain days. Handling missing values was done by the python pandas.

The interpolate () function fills NA values or missing values in the series or data frame. Rather than hard-coding the missing value, various interpolation, and convenient techniques could be used to fill the missing value.

C. Dataset Splitting

The dataset needed to be divided into training and testing sets to avoid any phenomenon such as overfitting. Besides, the size of the datasets as well as the train/test split ratios can significantly impact the model output, thus, affecting classification performance [50]. For example, if there are patterns in the training and testing set that do not exist in real-world data, the model performs poorly even though it cannot be seen in the performance evaluation. Dataset splitting is a practice considered indispensable and highly necessary to eliminate or reduce bias in training data for prediction models [51].

Based on the Rantau Panjang river dataset, 80% were training data, and the remaining 20% were testing data that would be optimally splitting the training and testing the dataset.

D. Data Transformation

Normalization is a scaling, mapping technique, or pre-processing stage. For prediction or forecasting purposes, it can be useful when it distinguishes a new range from an existing one.

Normalization is a transformation process that utilises a standard scale to produce numerically and comparably input data. After collecting input data, the data should perform some pre-processing to make it worthwhile for decision modelling [52]. As previously stated, this pre-processing should take three critical factors into account: 1) removed missing values from the data; 2) converted all non-numeric data to numerical data to allow for normalization (standardisation); 3) Determined how to select a suitable normalization technique to ensure a standard scale, appropriate modelling representation (benefit or cost criteria), and aggregation comparability to obtain alternative ratings.

After the data cleaning process, river stage and rainfall data underwent a min-max scaler to get normalized data. Min-max scaler scaled the data between the minimum and maximum value of the data that ended up ranging between 0 and 1. Another function for normalizing the data was to speed up the learning time and performance of the model. The data were scaled down to a range between [0, 1] or [-1, 1]. The method's equation of [14] can be referred to as (12):

$$a_{\text{norm}} = \frac{(\text{ high } - \text{ low })*(a-minA)}{maxA-minA} \tag{12}$$

min A is the smallest value, and max A is the largest value of attribute A.

E. Model Evaluation

1) MAE stands for Mean Absolute Error, which is

$$\frac{1}{n}\sum_1^n |y_i - \hat{y}_i| \tag{13}$$

Outliers are given less weight in this method, which is not sensitive to outliers. The equation of [53] can be referred to as (13).

2) MAPE stands for Mean Absolute Percentage Error, which is

$$\frac{1}{n}\sum_1^n |y_i - \hat{y}_i| \tag{14}$$

MAE is similar, but true observation is used to normalise it. The disadvantage is that this metric becomes problematic when true observation is zero. The equation of [53] can be referred to as (14).

3) MSE stands for Mean Squared Error, which is

$$\frac{1}{n}\sum_{i=1}^{\pi} \left(Y_i - \hat{Y}_i\right)^2 \tag{15}$$

MSE is a combination of variance of the prediction and measurement of bias, i.e., MSE = Bias² + variance. The equation of [53] can be referred to as (15).

4) RMSE stands for Root Mean Squared Error, which is

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2} \tag{16}$$

It measures the standard deviation of residuals. The equation of [53] can be referred to as (16).

5) R2 stands for coefficient of determination, which is

$$1-\frac{\sum_{i=1}^{n}(\hat{y}_i-y_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y}_i)^2} \quad (17)$$

Representing the coefficient indicates how well the values fit in comparison to the original values. The equation of [54] can be referred to as (17).

V. Results and Discussions

The results between the proposed models and original models that share the same hyperparameter setting were compared as shown in Table III.

In the present study, the proposed models were evaluated and compared with the original RNN model proposed by Hochreiter and Schmidhuber [55], the LSTM model proposed by Rumelhart & McClelland [56] and the BI-LSTM model proposed by Graves & Schmidhuber [57] to get a better understanding of whether the proposed models could produce better results [58][59][60]. The proposed models had an extra layer called layer normalization and one activation function is known as Leaky ReLU. In contrast, the original models had a standard layer and used a sigmoid as its activation function. Each deep learning model needs to be compared to the proposed and original models to ascertain which deep learning models perform better [61][62].

TABLE III. Comparison Result

Models	**MSE**	**MAE**	**RSME**	**MAPE (%)**	**R2**	**Training Time**
DRNN + LN + Leaky ReLU	0.107	0.233	0.327	4.293	0.946	1.08 minute
DRNN	0.118	0.242	0.343	4.344	0.94	0.38 second
LSTM + LN + Leaky ReLU	0.125	0.231	0.353	4.117	0.936	1.33 minute
LSTM	0.122	0.236	0.349	4.296	0.931	1.32 minute
BI-LSTM + LN + Leaky ReLU	0.131	0.258	0.362	4.751	0.933	3.33 minute
BI-LSTM	0.123	0.243	0.351	4.432	0.937	2.39 minute

Table III shows that the proposed models produced a better accuracy result: for example, the Deep Recurrent Neural Network (DRNN) + LN + Leaky ReLU model produced the lowest MSE among other models whereas the BI-LSTM + LN + Leaky ReLU model produced the highest MSE. For MAE, the LSTM + LN + Leaky ReLU model produced the lowest error and the BI-LSTM + LN + Leaky ReLU model produced the highest error among other models. For RSME, the DRNN + LN + Leaky ReLU model produced the lowest error whereas the BI-LSTM + LN + Leaky ReLU model produced the highest error. For MAPE, the interpretation was in percentage mode and the lower is better, where the LSTM + LN + Leaky ReLU model produced the lowest MAPE, and the BI-LSTM + LN + Leaky ReLU model produced the highest MAPE. For R2, the higher the interpretation is better, in which the DRNN + LN + Leaky ReLU model produced the highest R2 whereas the LSTM model produced the lowest R2.

Comparing the proposed models with the original models, the DRNN + LN + Leaky ReLU model produced a low error in terms of MSE, MAE, RSME, MAPE and R2 which indicated that the proposed models were good in making a prediction, but the drawback was that it took longer time to train the data. For the LSTM model, the LSTM + LN + Leaky ReLU model produced the lowest error in terms of MAE, MAPE and R2 which was slightly better than the LSTM model. For the BI-LSTM model, the BI-LSTM produced better results compared to the BI-LSTM + LN + Leaky ReLU model in terms of MSE, MAE, RSME, MAPE, R2 and training time. The proposed model could not work well with the BI-LSTM model.

Additionally, the BI-LSTM model required more training time compared to the other models. In addition to performance evaluation, training time must also be taken into account. For instance, it is clear that the proposed models needed more training time than the original model because it had an additional layer called the normalization layer.

In literature, the LSTM model is regarded as the best performer compared to other models developed with dependent variables. However, in this case, the DRNN model performance is better compared to other models. This result can be seen in Table III, in which the DRNN + LN + Leaky ReLU model outperform other models in terms of the MSE, RSME, R2 and training time. The LSTM model shows the second best among other deep learning models with the lowest MAE and MAPE with the LSTM + LN + Leaky ReLU model. The BI-LSTM model shows the lowest accuracy among other models, one thing needs to be highlighted the result has also shown that the accuracy among the models is about the same, the differences are just slight, and there is a considerable gap between them. The DRNN model is the first place in accuracy, followed by the LSTM model and the lowest is the BI-LSTM model. Because the LSTM framework uses backpropagation and a gate to train the model, it takes more time to train than the DRNN model, which uses sequential while training the data and has no backpropagation and gate in the architecture. The DRNN model ranks lowest in terms of training time, while the LSTM model comes in second. Contrarily, the BI-LSTM model necessitates that the training data move in both past and forward directions to train the data, which is why the BI-LSTM model takes longer to train the data.

Based on the result in Table III, only the proposed models with layer normalization and Leaky ReLU are deemed suitable for flood prediction with minimal missing value in the data usage which results in the lowest minimum error and good accuracy. The missing value in the data can be filled by using the linear interpolation method to get the best possible clean data. The authorities may use these models as an alternative to anticipate flooding and make enough preparations prior to its occurrence.

VI. Conclusion

In conclusion, the DRNN model performs relatively well compared to the LSTM and BI-LSTM models with the used dataset. From the literature, the LSTM architecture needs requirements for backpropagation and a gate to train the model. Therefore, the LSTM model is marginally more complicated than the DRNN model. Meanwhile, the BI-LSTM model performs with somewhat lower accuracy but is still able to deliver a good outcome. Additionally, the BI-LSTM model requires the training data to move backwards and forward in both directions which increases the training time needed. Even though the performance of proposed models performs well, there are still many improvements that can be made using deep learning approaches.

VII. Future Work

For future work, it is suggested that additional experiments be conducted by combining a statistic model with the pre-processing models to ascertain how the combined model performs. Currently, these models produce a good accuracy for one day ahead but for future work these models need to be tuned to produce a good accuracy for multi-days ahead.

Acknowledgment

The authors are grateful to Universiti Teknikal Malaysia Melaka for the financial support through the university's short-term grant PJP/2020/FTMK/PP/S01802.

References

[1] T. Ashizawa, N. Sudo, and H. Yamamoto, "How Do Floods Affect the Economy? An Empirical Analysis using Japanese Flood Data," 2022.

[2] T. Tiggeloven et al., "Global-scale benefit-cost analysis of coastal flood adaptation to different flood risk drivers using structural measures," Nat. Hazards Earth Syst. Sci., vol. 20, no. 4, pp. 1025–1044, 2020, doi: 10.5194/nhess-20-1025-2020.

[3] M. S. M. Shaari, M. Z. Abd Karim, and B. Hasan-Basri, "Does flood disaster lessen GDP growth? Evidence from Malaysia's manufacturing and agricultural sectors," Malaysian J. Econ. Stud., vol. 54, no. 1, pp. 61–81, 2017, doi: 10.22452/mjes.vol54no1.4.

[4] K. A. Oladapo, S. A. Idowu, Y. . Adekunle, and F. . Ayankoya, "Categorization of Conditioning Variables for Pluvial Flood Risk Assessment," Int. J. Sci. Eng. Res., vol. 11, no. 8, pp. 355–368, 2020.

[5] S. F. Zakaria, R. M. Zin, I. Mohamad, S. Balubaid, S. H. Mydin, and E. M. R. Mdr, "The development of flood map in Malaysia," AIP Conf. Proc., vol. 1903, no. November, 2017, doi: 10.1063/1.5011632.

[6] F. Nurashikin Sungip et al., "The Impact of Monsoon Flood Phenomenon on Tourism Sector in Kelantan, Malaysia: A Review," Int. J. Eng. Technol., vol. 7, no. 4.34, p. 37, 2018, doi: 10.14419/ijet.v7i4.34.23577.

[7] N. S. Romali, Z. Yusop, M. Sulaiman, and Z. Ismail, "Flood risk assessment: A review of flood damage estimation model for Malaysia," J. Teknol., vol. 80, no. 3, pp. 145–153, 2018, doi: 10.11113/jt.v80.11189.

[8] M. Zerara, "Machine Learning and Statistical Methods for Time Series Forecasting : a Case Machine Learning and Statistical Methods for Time Series Forecasting : a Case Study with Water Demand," no. March, 2021, doi: 10.36227/techrxiv.15019698.v1.

[9] A. Pant and R. S. Rajput, "Time Series Analysis of Gold Price Using R," no. January, 2018.

[10] M. Murat, I. Malinowska, M. Gos, and J. Krzyszczak, "Forecasting daily meteorological time series using ARIMA and regression models," Int. Agrophysics, vol. 32, no. 2, pp. 253–264, 2018, doi: 10.1515/intag-2017-0007.

[11] R. Yasrab, J. Zhang, P. Smyth, and M. P. Pound, "Predicting plant growth from time-series data using deep learning," Remote Sens., vol. 13, no. 3, pp. 1–17, 2021, doi: 10.3390/rs13030331.

[12] S. Joshi, "Time Series Analysis and Forecasting of the US Housing Starts using Econometric and Machine Learning Model," no. May, 2019.

[13] A. T. Jebb and L. Tay, Introduction to Time Series Analysis for Organizational Research: Methods for Longitudinal Analyses, vol. 20, no. 1. 2017.

[14] S. Bhanja and A. Das, "Deep Neural Network for Multivariate Time-Series Forecasting," Adv. Intell. Syst. Comput., vol. 1255, no. April, pp. 267–277, 2021, doi: 10.1007/978-981-15-7834-2_25.

[15] W. Wu, R. Emerton, Q. Duan, A. W. Wood, F. Wetterhall, and D. E. Robertson, "Ensemble flood forecasting: Current status and future opportunities," WIREs Water, vol. 7, no. 3, 2020, doi: 10.1002/wat2.1432.

[16] Z. Shen, Y. Zhang, J. Lu, J. Xu, and G. Xiao, "A novel time series forecasting model with deep learning," Neurocomputing, vol. 396, no. 0925–2313, pp. 302–313, 2020, doi: https://doi.org/10.1016/j.neucom.2018.12.084.

[17] C. L. Jun, Z. S. Mohamed, A. L. S. Peik, S. F. M. Razali, and S. Sharil, "Flood forecasting model using empirical method for a small catchment area," J. Eng. Sci. Technol., vol. 11, no. 5, pp. 666–672, 2016.

[18] A. Faruq, H. P. Arsa, S. F. M. Hussein, C. M. C. Razali, A. Marto, and S. S. Abdullah, "Deep Learning-Based Forecast and Warning of Floods in Klang River, Malaysia," Ing. des Syst. d'Information, vol. 25, no. 3, pp. 365–370, 2020, doi: 10.18280/isi.250311.

[19] S. A. Asmai, Z. Z. Abidin, H. Basiron, and S. Ahmad, "An intelligent crisis-mapping framework for flood prediction," Int. J. Recent Technol. Eng., vol. 8, no. 2 Special Issue 8, pp. 1304–1310, 2019, doi: 10.35940/ijrte.B1058.0882S819.

[20] S. Siami-Namini and A. S. Namin, "Forecasting Economics and Financial Time Series: ARIMA vs. LSTM," pp. 1–19, 2018.

[21] J. K. Jaiswal and R. Das, "Artificial neural network algorithms based nonlinear data analysis for forecasting in the finance sector," Int. J. Eng. Technol., vol. 7, no. 4, pp. 169–176, 2018, doi: 10.14419/ijet.v7i4.10.20829.

[22] A. Šiljić Tomić, D. Antanasijević, M. Ristić, A. Perić-Grujić, and V. Pocajt, "A linear and non-linear polynomial neural network modeling of dissolved oxygen content in surface water: Inter- and extrapolation performance with inputs' significance analysis," Sci. Total Environ., vol. 610–611, pp. 1038–1046, 2018, doi: 10.1016/j.scitotenv.2017.08.192.

[23] Y. F. Zhang, P. Fitch, and P. J. Thorburn, "Predicting the trend of dissolved oxygen based on the PCA-RNN model," Water (Switzerland), vol. 12, no. 2, 2020, doi: 10.3390/w12020585.

[24] M. Coto-Jiménez, "Experimental study on transfer learning in denoising autoencoders for speech enhancement," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12088 LNCS, pp. 307–317, 2020, doi: 10.1007/978-3-030-49076-8_29.

[25] A. Z. Dhunny, R. H. Seebocus, Z. Allam, M. Y. Chuttur, M. Eltahan, and H. Mehta, "Flood Prediction using Artificial Neural Networks: Empirical Evidence from Mauritius as a Case Study," Knowl. Eng. Data Sci., vol. 3, no. 1, pp. 1–10, 2020, doi: 10.17977/um018v3i12020p1-10.

[26] V. Kumar and M. L., "Deep Learning as a Frontier of Machine Learning: A Review," Int. J. Comput. Appl., vol. 182, no. 1, pp. 22–30, 2018, doi: 10.5120/ijca2018917433.

[27] A. Mathew, P. Amudha, and S. Sivakumari, "Deep learning techniques: an overview," Adv. Intell. Syst. Comput., vol. 1141, no. January, pp. 599–608, 2021, doi: 10.1007/978-981-15-3383-9_54.

[28] R. Vargas, Rocio, Mosavi, Amir, & Ruiz, "Deep Learning : a Review Deep Learning : a Review," Adv. Intell. Syst. Comput., no. July, 2017.

[29] J. Fu, J. Chu, P. Guo, and Z. Chen, "Condition Monitoring of Wind Turbine Gearbox Bearing Based on Deep Learning Model," IEEE Access, vol. 7, no. April, pp. 57078–57087, 2019, doi: 10.1109/ACCESS.2019.2912621.

[30] L. Hu, J. Zhang, Y. Xiang, and W. Wang, "Neural Networks-Based Aerodynamic Data Modeling: A Comprehensive Review," IEEE Access, vol. 8, pp. 90805–90823, 2020, doi: 10.1109/ACCESS.2020.2993562.

[31] G. Chen, "A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation," pp. 1–10, 2016.

[32] H. H. Hasan, S. F. M. Razali, A. Z. I. A. Zaki, and F.-3. pdfu. M. Hamzah, "Integrated hydrological-hydraulic model for flood simulation in tropical urban catchment," Sustain., vol. 11, no. 23, 2019, doi: 10.3390/su11236700.

[33] B. Biswal, "Hydrological modelling: scale issues and philosophical approaches," no. May, pp. 36–50, 2019.

[34] E. Brown et al., "Methods and tools to support real time risk-based flood forecasting - A UK pilot application," E3S Web Conf., vol. 7, no. October, 2016, doi: 10.1051/e3sconf/20160718019.

[35] M. Bayat and O. Tavakkoli, "Application of machine learning in flood forecasting," Futur. Technol., vol. 1, no. 1, pp. 1–6, 2020, doi: 10.55670/fpll.futech.1.1.1.

[36] X. H. Le, H. V. Ho, G. Lee, and S. Jung, "Application of Long Short-Term Memory (LSTM) neural network for flood forecasting," Water (Switzerland), vol. 11, no. 7, 2019, doi: 10.3390/w11071387.

[37] P. Goymann, D. Herrling, and A. Rausch, "Flood Prediction through Artificial Neural Networks A case study in Goslar , Lower Saxony," no. c, pp. 56–62, 2019.

[38] V. Gude, S. Corns, and S. Long, "Flood Prediction and Uncertainty Estimation Using Deep Learning," Water (Switzerland), vol. 12, no. 3, 2020, doi: 10.3390/w12030884.

[39] T. Song, W. Ding, J. Wu, H. Liu, H. Zhou, and J. Chu, "Flash flood forecasting based on long short-term memory networks," Water (Switzerland), vol. 12, no. 1, 2020, doi: 10.3390/w12010109.

[40] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," 2018.

[41] P. Lara-Benítez, M. Carranza-García, J. M. Luna-Romera, and J. C. Riquelme, "Temporal convolutional networks applied to energy-related time series forecasting," Appl. Sci., vol. 10, no. 7, 2020, doi: 10.3390/app10072322.

[42] X. Bi and L. Wang, "Performing Weakly Supervised Retail Instance Segmentation via Region Normalization," IEEE Access, vol. 9, pp. 67761–67775, 2021, doi: 10.1109/ACCESS.2021.3077031.

[43] J. Xu, X. Sun, Z. Zhang, G. Zhao, and J. Lin, "Understanding and improving layer normalization," Adv. Neural Inf. Process. Syst., vol. 32, no. November, 2019.

[44] G. Renith and A. Senthilselvi, "Accuracy improvement in diabetic retinopathy detection using dlia," J. Adv. Res. Dyn. Control Syst., vol. 12, no. 4, pp. 133–149, 2020, doi: 10.5373/JARDCS/V12I4/20201426.

[45] S. Nasiri, J. Helsper, M. Jung, and M. Fathi, "DePicT Melanoma Deep-CLASS: A deep convolutional neural networks approach to classify skin lesion images," BMC Bioinformatics, vol. 21, no. Suppl 2, pp. 1–14, 2020, doi: 10.1186/s12859-020-3351-y.

[46] N. A. Mohamd Hanan et al., "A GIS-Based Flood Vulnerability Assessment in Pasir Mas, Kelantan," IOP Conf. Ser. Earth Environ. Sci., vol. 549, no. 1, 2020, doi: 10.1088/1755-1315/549/1/012004.

[47] N. Ahmad Zamree, N. A. Said, and S. Sibly, "Hospital Disaster Preparedness: A Model for Hospital Disaster Preparedness Based on 2014 Flood in Kelantan," Educ. Med. J., vol. 10, no. 4, pp. 69–80, 2019, doi: 10.21315/eimj2018.10.4.7.

[48] F. Ridzuan and W. M. N. Wan Zainon, "A review on data cleansing methods for big data," Procedia Comput. Sci., vol. 161, pp. 731–738, 2019, doi: 10.1016/j.procs.2019.11.177.

[49] P. Xu and Y. Jia, "SNR improvement based on piecewise linear interpolation," J. Electr. Eng., vol. 72, no. 5, pp. 348–351, 2021, doi: 10.2478/jee-2021-0049.

[50] A. Rácz, D. Bajusz, and K. Héberger, "Effect of dataset size and train/test split ratios in qsar/qspr multiclass classification," Molecules, vol. 26, no. 4, pp. 1–16, 2021, doi: 10.3390/molecules26041111.

[51] I. Muraina, "IDEAL DATASET SPLITTING RATIOS IN MACHINE LEARNING ALGORITHMS :," no. February, 2022.

[52] B. Etzkorn, "Data Normalization and Standardization," pp. 1–3, 2018.

[53] L. Chen, X. Yang, C. Sun, Y. Wang, D. Xu, and C. Zhou, "Feed intake prediction model for group fish using the MEA-BP neural network in intensive aquaculture," Inf. Process. Agric., vol. 7, no. 2, pp. 261–271, 2020, doi: 10.1016/j.inpa.2019.09.001.

[54] B. Panay, N. Baloian, J. Pino, S. Peñafiel, H. Sanson, and N. Bersano, "Predicting Health Care Costs Using Evidence Regression," p. 74, 2019, doi: 10.3390/proceedings2019031074.

[55] W. Feng, N. Guan, Y. Li, X. Zhang, and Z. Luo, "Audio visual speech recognition with multimodal recurrent neural networks," Proc. Int. Jt. Conf. Neural Networks, vol. 2017-May, no. May, pp. 681–688, 2017, doi: 10.1109/IJCNN.2017.7965918.

[56] Z. Yu and G. Liu, "Sliced recurrent neural networks," COLING 2018 - 27th Int. Conf. Comput. Linguist. Proc., pp. 2953–2964, 2018.

[57] T. Zhu, J. Shen, and F. Sun, "Long Short-Term Memory-based simulation study of river happiness evaluation – A case study of Jiangsu section of Huaihe River Basin in China," Heliyon, vol. 8, no. 9, p. e10550, 2022, doi: 10.1016/j.heliyon.2022.e10550.

[58] Ammar Ashraf Narul Akhla, Thong Chee Ling, Abdul Samad Shibghatullah, Chit Su Mon, Aswani Kumar Cherukuri, Chaw Lee Yen and Lee Chiw Yi,2022, "Impact of Real-Time Information for Travellers: A Systematic Review", Journal of Information & Knowledge Management, vol. 21, no.4,pp. 2250065-1-2250065-21, 2022.

[59] Susanto, I.C., Subaramaniam, K., Shibghatullah, A.S.B.,"Gesturenomy: Touchless Restaurant Menu Using Hand Gesture Recognition", Proceedings of International Conference on Artificial Life and Robotics, pp.229-236, 2022.

[60] Meng, W.Y., Shibghatullah, A.S.B., Subaramaniam, K.,"Smart Tourism Guide Application Using Location-Based Services-Go.Travel", Proceedings of International Conference on Artificial Life and Robotics, pp. 219-228, 2022.

[61] Shibghatullah, A.S., Jalil, A., Wahab, M.H.A., Soon, J.N.P., Subaramaniam, K., Eldabi, T.,"Vehicle Tracking Application Based on Real Time Traffic", International Journal of Electrical and Electronic Engineering and Telecommunications, vol. 11, no. 1,pp. 67-73, 2022

[62] Haini, M.S.B., Mon, C.S., Shibghatullah, A.S.B., Jalil, A.B., Subaramaniam, K.A.P., Hussin, A.A.A., "An investigation into requirement of mobile app for apartment residents", International Journal on Advanced Science, Engineering and Information Technology, vol. 9, no. 6, pp. 1841-1848, 2019.