Comparison study between selected techniques of (ML, SVM and Deep Learning) regarding prediction of Flooding in Eastof Iraq

*1Hassan M. Idan,*2 Dr. Karim Q. Hussein
*1 Computer Science, Iraqi commission for computer and informatics, Informatics Institute for Postgraduate Studies, Baghdad, Iraq. Hassanprince807@gmail.com, ms201910541@iips.icci.edu.iq
*2Assist. Prof. /Computer Science Dept. /Faculty of Science/ Mustansiryha University / Baghdad, Iraq. Karimzzm@yahoo.com, karim.q.h@uomustansiriyha.edu.iq

Abstract

East of Iraq Regions such as wasit city is one of the flood affected Regions, cussed by The torrents coming from Iran and the rain water that causes great danger to the people who live near the Iraqi-Iranian border, A good amount of work carried out by machine learning (ML) techniques and deep learning in the past for flood occurrence based on rainfall, humidity, temperature, water flow, water level etc. The problem is that no one has attempted to predict the likelihood of a flood based on temperature and rainfall intensity.

Therefore, we use deep learning models such as Convolutional neural network(CNN), Recurrent Neural Network(RNN), Multi-layer Perceptron (MLP) and machine learning model such as Support vector machine(SVM), k-Nearest Neighbors(KNN), Decision Tree(DT),Random forest(RF) and Logistic regression(LR) for predicting the occurrence of flood based on temperature and rainfall intensity and the results were compared between the deep learning models and machine learning models them in terms of accuracy, recall, precision and F1 Score.

The results indicate that the CNN algorithm of deep learning and KNN algorithm of machine learning and can be efficiently used for flood forecasting with highest accuracy based on rainfall parameters, Amount of running river water and temperature before flood occurrence.

Keywords: Deep learning, Machine learning,Flood forecasting,flood prediction,forecasting

1. Introduction

One of the most important disasters that occur quickly and widely is floods that cause great damage to human life that may reach death, as well as agriculture and the economy, Therefore governments must develop reliable and accurate maps of areas with high risk knowledge and ongoing plans for flood risks and focus on prevention, preparedness, and protection of individuals[1].Flood forecasting models are of great importance for predicting floods before they occur, as well as reducing risks to facilitate the process of managing water resources, suggesting policies and analysing data, and evacuation modelling[2],However, due to the changing nature of climatic conditions, flood lead-time and occurrence location forecast is inherently complicated. As a result, today's primary flood prediction models are primarily data-driven and rely on a number of simplified assumptions[3].Although climate change has consequences as it causes an increase in the possibility of floods, such as an increase in the amount of precipitation if torrential rains come from countries, or snow melting in some countries, because people began to live near rivers, the danger is greater and safety measures and precautions must be provided to reduce Flood-related deaths and other damages[4]

Forecasting floods at the same time is a difficult task, because it needs to identify risks and determine the possibility of floods, because it depends on measurements of climate change from rain, heat and wind intensity, and any error in measurements or uncertainty causes errors in prediction and can lead to increased damage[5].The areas in central Iraq adjacent to the Iranian borders are vulnerable to flooding due to the quantities of water coming from Iran due to torrential rains, heavy rains, or any emergency circumstance that poses a threat to the residents of these areas. Computational methods like as neural networks have been widely used to predict flood in a river's endangered region and its influence outside of the specified area: for example, the upstream river flow or discharge is extremely useful in locating downstream flows that are not equipped or lack measurements[6].Floods, whether natural or flash floods, are typically the consequence of excessive rainfall. For example, the devastating flood occurrence in Malaysia in 2014[7]taught us the need of having a flood prediction system in place to monitor, anticipate, and detect flood events.

It is critical to give a flood warning as soon as possible in order to minimize such losses. As a result, water level forecasting is critical for predicting future floods. Agriculture, plants, domestics, and industrial and commercial sectors all benefit from water level forecast[8].

The aim of this research paper is to develop a predictive modelling Standard Process for Data Mining methodology by using Support vector machine(SVM) and other Machine Learning (ML) techniques such as Decision Tree (DT), k-Nearest Neighbours (KNN) and random forest and logistic regression Deep Learning (DL) techniques such as convolutional neural network (CNN), recurrent neural networks(RNN) and multilayer perceptron(MLP) and for flood prediction in East of Iraq and then compare the result of all models by the evaluation metrics

The remaining of this paper is organized as follows. Section 2 reviews all works related to techniques used for flood risk prediction, Section 3 presents the concept of Prediction methodology and prediction techniques, Section 4 Provides a description of the database being used, Section 5 presents the results for all models used In from its beginning of dataset pre-processing to evaluate the results and Finally Section 6 concludes with some directions for future work.

2. Related Work

- Razali et al.[9]The study aims to develop a flood forecasting model by using machine learning techniques such as Bayesian (BN) and other machine learning (ML) techniques such as Decision Tree (DT), k-Nearest Neighbors (KNN) and Support Vector Machine (SVM) to predict flood risk. By reaching 99.92 percent accuracy, the DT technique outperformed the others**.**

- Zehra.[10]Changing the behavior of river water can cause floods This paper suggested a use Non-linear (NARX) and Support Vector Machine (SVM) are machine learning algorithms to predict changes in river water levels and thus the probability of flood detection. Precipitation amount, river inflow, peak gust, seasonal flow, flood frequency, and other important flood forecast factors are used by both algorithms. Machine-learning algorithms are excellent in predicting floods because they can use data from a variety of sources and categorize and regress it into flood and non-flood classes. Based on the comparative synthesis, conclude it is determined that statistical techniques combined with NARX may give extremely accurate and promising flood forecasting outcomes.

- Sankaranarayanan et al.[11]This study discusses how to predict floods based on temperature and precipitation intensity, where a deep learning model was used with other machine learning models (Support Vector Machine (SVM), K nearest neighbor (KNN) and Naïve Bayes) and compared in terms of accuracy and error terms. The findings show that the deep neural network can be effectively utilized for flood forecasting with the best accuracy based on monsoon factors only prior to the onset of floods.

- Kumar et al.[12]This paper demonstrates to the use of ANNs to predict water flows. Two different networks were used namely the feed forward network and the recurrent neural network, have been chosen to predict the flow of the river in India to overcome the floods. A comparison of the two networks found that the recurrent neural networks outperformed the feed forward networks. Furthermore, the recurrent neural networks had a smaller design and required less training time. Both single step ahead and multiple step ahead forecasting yielded superior results using the recurrent neural network. As a result, recurrent neural networks are recommended as a technique for predicting flood flow.

- Dtissibe et al.[13]This paper describes how floods have become a major threat to the environment and economy of countries causing loss of life and material damage. It is necessary to build a flood forecasting system. Physical methods for flood detection have proven to be limited and ineffective. Use machine learning tools because neural network systems are a good alternative. The focus of attention was the performance of the models and minimum prediction errors. Multiple layers were used to design the flood forecasting model. The model has been tested based on experiments, and the results show the effectiveness of the proposal with good predictive ability.

- Kunverji et al.[14]This paper discusses the importance of a flood forecasting system, as floods have become a catastrophic event and the lack of a flood forecasting system has resulted in huge losses. The goal of this study is to developing the most effective flood forecasting model. AI calculations and a productive and precise flood forecasting system give all of the support require to Residents and the government. Then building a decision tree model. This model performs different computations on datasets with a high level of precision. The model uses artificial intelligence to predict floods and send notifications to local officials and municipalities. Decision tree, random forest, and gradient boosting of three machine learning methods used for comparison. This model focuses on increasing prediction rates by utilizing more complex data and a high-level algorithm. We build models for machine learning and deep learning, compare the results of the models and choose the best to predict floods.

3. Prediction

Prediction is a statistical technique that uses machine learning, deep learning, and data mining to predict possible future outcomes with the help of historical and current data.Where the work of modeling is by analyzing current and historical data and projecting what it has learned onto the model that was created to predict possible outcomes.Predictive modeling can be used to predict everything. Predictive models work so quickly and can be used to forecasting such as TV ratings, disease forecasting, natural accidents, credit risk, corporate earnings, and online betting risks [15].Can see in the below figure 1 basic flow for building model for Prediction. Inour thesis, machine learning and deep learning supervised algorithms were used to predict floods, and the algorithm results for both techniques were compared.

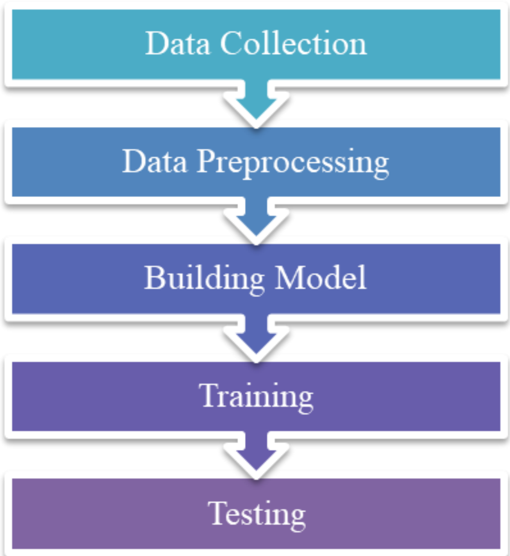


Figure 1 Basic flow for building model forPrediction

3.1. ML methods in flood prediction

Machine learning is an evolving field of computing algorithms that aim to mimic human intelligence by learning from their surroundings, they are the best thing to analysis big data, and Machine learning techniques have been used in a variety of disciplines, including computer vision, spacecraft engineering, as well as biomedical and medical applications[16]. We will use some of ML algorithms to prediction flooding such as k-Nearest Neighbors, Logistic regression, Decision Tree, Random Forest and Support Vector Machine.

3.2. Deep learning methods in flood prediction

The term Deep Learning or Deep Neural Network refers to Artificial Neural Networks (ANN) with multi layers. Over the last few decades, it has been considered to be one of the most powerful tools, and has become very popular in the literature as it is able to handle a huge amount of data. The interest in having deeper hidden layers has recently begun to surpass classical methods performance in different fields; especially in pattern recognition[17]. One of the most popular deep neural networks is the Convolutional neural networks (CNN), Recurrent neural networks (RNN)and Multilayer perceptron (MLP).

4. Data Set

Represents the phase of collecting data we used a set of databases for Iraq's climate from the NASA website from 1990 to 2020 and the database includes all the details of the climate in Iraq and for all months of the year of temperatures and rainfall and their rates. To determine the areas covered by water when flooding. Below figure 2 show the dataset.

	A
1	SUBDIVISION,YEAR,Time,Flood_possibility ,Amount_of_water_m,Area_covered_by_water,propagation_speed,JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC, ANNUAL RAINFALL,FLOODS
2	AL-kut,1980,1:00 AM,65.684998,66.842499,65.307503,66.517502,0,0.9,20.8,114.8,105.3,745.9,754,438.1,139.5,282.3,162.3,39.5,2803.4,NO
3	AL-kut,1981,2:00 AM,67.175003,67.517502,66.175003,66.997498,7,6.8,28.5,75.9,166.3,912.4,489.8,495.6,376.6,265,138.6,43.3,3005.9,YES
4	AL-kut,1982,3:00 AM,67.077499,68.425003,66.457497,68.3125,0.7,0.1,21.9,60.4,148.2,612.2,511.5,495,70.6,164.4,127.5,10.8,2223.3,NO
5	AL-kut,1983,4:00 AM,70,72.0625,69.512497,71.762497,0.2,1.5,0.9,13.1,76,322.8,583.2,579.9,421.1,136.2,116.5,69.1,2320.3,NO
6	AL-kut,1984,5:00 AM,70.599998,71.582497,70.157501,71.107498,36.8,60,95.3,162.1,84.6,842.6,653.6,284.4,171.1,286,67.7,18,2762.1,NO
7	AL-kut,1985,6:00 AM,71.845001,72.050003,70.587502,71.672501,61.2,6.1,29.3,66.6,254.2,828.7,388.9,315.3,117.6,204,74.9,44,2390.5,NO
8	AL-kut,1986,7:00 AM,71.172501,71.737503,69.214996,70.699997,5.6,18.7,11.2,63.1,126.7,597.9,324.8,340.3,235.4,165.5,194.7,9.5,2093.2,NO
9	AL-kut,1987,8:00 AM,69.487503,70.419998,69.212502,69.232498,0.6,0.8,4.3,57.2,108.3,572.6,221.396.6,157,272.1,216,131.1,2137.6,NO

Figure 2 Dataset

5. Result and destination:

5.1 Data Pre-processing

Figure 3 shows the original data that it contains unimportant values and does not affect our work and null value Text values that we converted to numerical values as shown inFigure 4.

```
      Country      Month  Year  Rainfall - (MM)  Temperature - (Celsius) FLOODS
0        Iraq    January  1901             32.9                      7.9    Yes
1        Iraq   February  1901             28.5                     11.9    Yes
2        Iraq      March  1901             30.1                     17.0    Yes
3        Iraq      April  1901             27.3                     21.6     No
4        Iraq        May  1901             12.1                     26.2     No
...       ...        ...   ...              ...                      ...    ...
1435     Iraq     August  2020              0.6                     33.7     No
1436     Iraq  September  2020              0.1                     31.8     No
1437     Iraq    October  2020              3.5                     25.1     No
1438     Iraq   November  2020             33.7                     17.2     No
1439     Iraq   December  2020             22.4                     11.2    Yes

[1440 rows x 6 columns]
```

Figure 3 Original Dataset

	Country	Month	Year	Rainfall - (MM)	Temperature - (Celsius)	Month_No	FLOODS
0	Iraq	January	1901	32.9	7.9	1	1
1	Iraq	February	1901	28.5	11.9	2	1
2	Iraq	March	1901	30.1	17.0	3	1
3	Iraq	April	1901	27.3	21.6	4	0
4	Iraq	May	1901	12.1	26.2	5	0

Figure 4 Dataset after pre-processing

5.2 Features extraction

After Pre-processing we split the data to training and testing data, we do first split them to training data (70%) and testing data (30%), the figure show the training data without label values and the figure show testing data content only one value, Due to the presence of a varying value in the quantities, we made a normalization between (0-1) for the data to get rid of it see figure(5,6, 7).

	Rainfall - (MM)	Temperature - (Celsius)	Month_No
0	32.9	7.9	1
1	28.5	11.9	2
2	30.1	17.0	3
3	27.3	21.6	4
4	12.1	26.2	5

Figure 5 training data

```
1435    0
1436    0
1437    0
1438    0
1439    1
```

Figure 6 testing data

```
array([[0.49031297, 0.14417178, 0.        ],
       [0.4247392 , 0.26687117, 0.09090909],
       [0.4485842 , 0.42331288, 0.18181818],
       ...,
       [0.05216095, 0.67177914, 0.81818182],
       [0.50223547, 0.42944785, 0.90909091],
       [0.3338301 , 0.24539877, 1.        ]])
```

Figure 7 data after normalization

5.3 Machine learning models

5.3.1 k-Nearest Neighbors

5.3.1.1 Architecture

It is important to get the optimum value of K, so that the model can classify well, we tested the accuracy of the training and test data by increasing the value of the number of neighbors from 1 to 10 illustrated in the figure 8. After that, we select the best value for K.

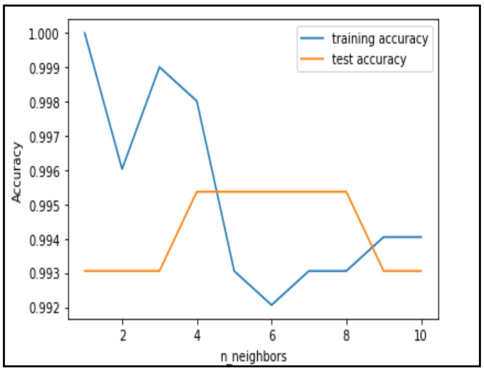


Figure 8Best number of neighbors

The figure 8 shows that the best value for k is number 9. Then we make the new value of K equal to 9. Then we note the accuracy value of the training and test data Accuracy of K-NN classifier on training set: 0.99, Accuracy of K-NN classifier on test set: 1.00 and prediction value to how accurate is our model is show in below Figure 9:

```
Accuracy Score:99.537037
Recall Score:100.000000
ROC score:99.686520
[[317   2]
 [  0 113]]
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       319
           1       0.98      1.00      0.99       113

    accuracy                           1.00       432
   macro avg       0.99      1.00      0.99       432
weighted avg       1.00      1.00      1.00       432
```

Figure 9KNN evaluation

5.3.2 Logistic regression

5.3.2.1 Architecture

We built a logistic regression model using a different measure of the C modulus where C is the inverse of the strength of regulation. It should be a positive float. As in support vector machines, smaller values define a stronger organization and we compare the final values as a result of the different C value.in figure 10, 11, 12 Show result change by valueC.

```
accuracy score:94.444444
recall score:84.955752
roc score:91.380697
[[312   7]
 [ 17  96]]
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       319
           1       0.98      1.00      0.99       113

    accuracy                           1.00       432
   macro avg       0.99      1.00      0.99       432
weighted avg       1.00      1.00      1.00       432
```

Figure 10LR evaluation(c=1)

```
accuracy score:94.907407
recall score:86.725664
roc score:92.265653
[[312   7]
 [ 15  98]]
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       319
           1       0.98      1.00      0.99       113

    accuracy                           1.00       432
   macro avg       0.99      1.00      0.99       432
weighted avg       1.00      1.00      1.00       432
```

Figure 11LR evaluation(c=0.01)

```
accuracy score:94.444444
recall score:84.955752
roc score:91.380697
[[312   7]
 [ 17  96]]
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       319
           1       0.98      1.00      0.99       113

    accuracy                           1.00       432
   macro avg       0.99      1.00      0.99       432
weighted avg       1.00      1.00      1.00       432
```

Figure 12LR evaluation(c=100)

After look to the above figure best result oflogistic regression model when the value of C=0.01.

5.3.3 Decision Tree

We built a Decision Tree model using a different parameters such as **max_depth**the maximum depth of the tree. If none, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples, **random_state** Controls the randomness of the estimator. The features are always randomly permuted at each split, even if splitter is set to "best",Where we in the first model assign value of max_depth equal to 3 and the random_state equal none then we got the result of Accuracy on training set: 1.000, Accuracy on test set: 0.993,And the second model assign value of max_depth equal to none and the random_state equal none then we got the result of Accuracy on training set: 1.000, Accuracy on test set: 0.993, The best model is the second as show in figure 13 below:

```
accuracy score:99.305556
recall score:100.000000
roc score:99.529781
[[316   3]
 [  0 113]]
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       319
           1       0.97      1.00      0.99       113

    accuracy                           0.99       432
   macro avg       0.99      1.00      0.99       432
weighted avg       0.99      0.99      0.99       432
```

Figure 13Decision Tree evaluation

5.3.4 Random Forest

Where talk about building Random Forest we design two model with change value of parameters of Random Forest such as n_estimatorsint, default The number of trees in the forest, max_depth, random state.First model have assign value to parameters (n_estimators=100, max_depth=100, random_state=0,the results were as follows Accuracy on training set: 1.000, Accuracy on test set: 0.995,Second model have assign value to parameters (n_estimators=200, max_depth=210, random_state=0, the results were as follows Accuracy on training set: 1.000, Accuracy on test set: 0.998, the figure 14 show the second model is the best :

```
accuracy score:99.768519
recall score:100.000000
roc score:99.843260
[[318   1]
 [  0 113]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       319
           1       0.99      1.00      1.00       113

    accuracy                           1.00       432
   macro avg       1.00      1.00      1.00       432
weighted avg       1.00      1.00      1.00       432
```

Figure 14Random Forest evaluation

5.3.5 Support Vector Machine

5.3.5.1 Architecture

Like as previous models, we built more than one SVM model with variable parameter values such as C Regularization parameter, kernel Specifies the kernel type to be used in the algorithm, First model assign value to C=1.0 and kernel =linear then we got the result of Accuracy on training set: 0.95, Accuracy on test set: 0.95, Second model assign value to C=1.0 and kernel = sigmoid then we got the result of Accuracy on training set: 0.770, Accuracy on test set: 0.780.Third model assign value to C=1.0 and kernel = rbf then we got the result of Accuracy on training set: 0.99, Accuracy on test set: 0.98,Fourth model assign value to C=12 and kernel = rbf then we got the result of Accuracy on training set: 1.00, Accuracy on test set: 0.99, when we look at the results we obtained, it becomes clear to us that the fourth model is the best, and the figure 15 shows the details:

```
accuracy score:98.611111
recall score:97.345133
roc score:98.202347
[[316   3]
 [  3 110]]
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       319
           1       0.97      0.97      0.97       113

    accuracy                           0.99       432
   macro avg       0.98      0.98      0.98       432
weighted avg       0.99      0.99      0.99       432
```

Figure 15SVM evaluation

5.3.6 Computingenvironment of machine learning algorithms

Typical training in this project was conducted on Windows 10 PC equipped with Intel core i7 CPU, NVIDIA 2GB GeForce 820M GPU. The models was developed using scikit-learn 0.23.1.

5.4 Deep learning model

5.4.1.1 CNN

5.4.1.2 Architecture

CNN model is designed with a VGG (Visual Geometry Set) model structure that has a first layer conv1d block with 64 filters, This layer creates a convolution kernel that is convolved with the layer input over a single spatial (or temporal) dimension to produce a tensor of outputs ,Next is Dense layer operation on the input and return the output,MaxPooling1D layer used basically for the purpose of reducing the size of the data and flatten layer that use to converting the data into a 1-dimensional array for inputting it to the next layer,The final model contains two layers is flatten,Dense, Each convolution layer contains 64 filters, ReLU activation, and Unified Core Configurator with the same padding to make sure the output function mappings are the same width and height and dropout layer The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting and final layer is Dense. The developed CNN architecture is shown in Figure (16-17).

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_1 (Conv1D)            (None, 2, 128)            384
_________________________________________________________________
dense_3 (Dense)              (None, 2, 128)            16512
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 1, 128)            0
_________________________________________________________________
flatten_2 (Flatten)          (None, 128)               0
_________________________________________________________________
dense_4 (Dense)              (None, 32)                4128
_________________________________________________________________
flatten_3 (Flatten)          (None, 32)                0
_________________________________________________________________
dropout_1 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_5 (Dense)              (None, 2)                 66
=================================================================
Total params: 21,090
Trainable params: 21,090
Non-trainable params: 0
```

Figure 16Details of CNN architecture and result output shape each layer

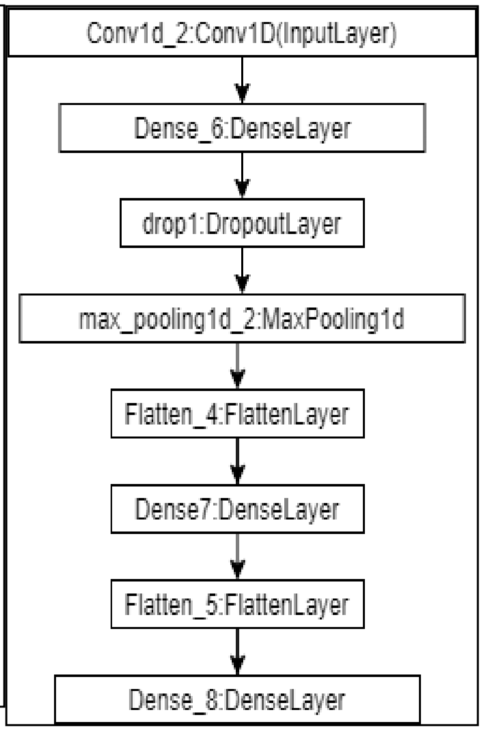


Figure 17Model architecture

5.4.1.3 Model Improvement and Training Procedure

We see the accuracy in the training and testing process in Figure 18, 19shows a representation of the loss in this process, it takes a few hours to complete this process.

Here it turns out that when the number of epochs is 32, batch size is10, we will obtain the highest verification accuracy (0.99%) corresponding to the lowest validation loss, so it is considered the optimal number of epochs.

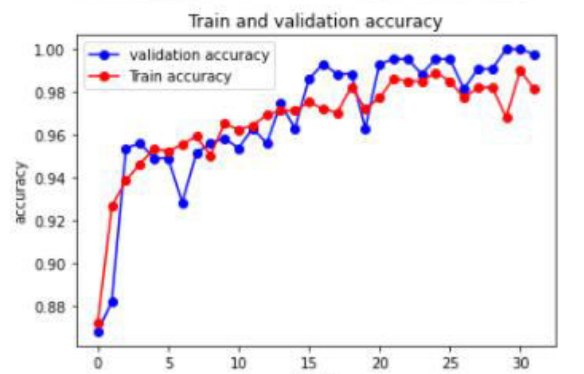


Figure 18accuracy of Training and validation process

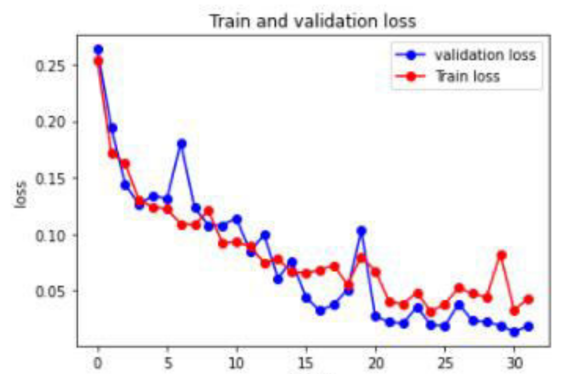


Figure 19loss of Training and validationprocess

```
Epoch 23/32
101/101 [==============================] - 0s 3ms/step - loss: 0.0316 - accuracy: 0.9883 - val_loss: 0.0209 - val_accuracy: 0.9
954
Epoch 24/32
101/101 [==============================] - 0s 3ms/step - loss: 0.0508 - accuracy: 0.9828 - val_loss: 0.0355 - val_accuracy: 0.9
884
Epoch 25/32
101/101 [==============================] - 0s 2ms/step - loss: 0.0377 - accuracy: 0.9860 - val_loss: 0.0196 - val_accuracy: 0.9
954
Epoch 26/32
101/101 [==============================] - 0s 2ms/step - loss: 0.0350 - accuracy: 0.9845 - val_loss: 0.0188 - val_accuracy: 0.9
954
Epoch 27/32
101/101 [==============================] - 0s 2ms/step - loss: 0.0474 - accuracy: 0.9722 - val_loss: 0.0381 - val_accuracy: 0.9
815
Epoch 28/32
101/101 [==============================] - 0s 2ms/step - loss: 0.0465 - accuracy: 0.9833 - val_loss: 0.0233 - val_accuracy: 0.9
907
Epoch 29/32
101/101 [==============================] - 0s 2ms/step - loss: 0.0433 - accuracy: 0.9817 - val_loss: 0.0226 - val_accuracy: 0.9
907
Epoch 30/32
101/101 [==============================] - 0s 2ms/step - loss: 0.1121 - accuracy: 0.9570 - val_loss: 0.0189 - val_accuracy: 1.0
000
Epoch 31/32
101/101 [==============================] - 0s 2ms/step - loss: 0.0319 - accuracy: 0.9899 - val_loss: 0.0139 - val_accuracy: 1.0
000
Epoch 32/32
101/101 [==============================] - 0s 2ms/step - loss: 0.0357 - accuracy: 0.9867 - val_loss: 0.0184 - val_accuracy: 0.9
977
32/32 [==============================] - 0s 2ms/step - loss: 0.0268 - accuracy: 0.9940
Train Loss: 0.026766089722514153  Train accuracy: 0.9940476417541504
14/14 [==============================] - 0s 1ms/step - loss: 0.0184 - accuracy: 0.9977
Test Loss: 0.018390916287899017  Test accuracy: 0.9976851940155029
```

Figure 20CNN Training process and testing process results

When looking at the Figure 20 that represents the results of training the model and figure 21 below shows the correct cases for the prediction.

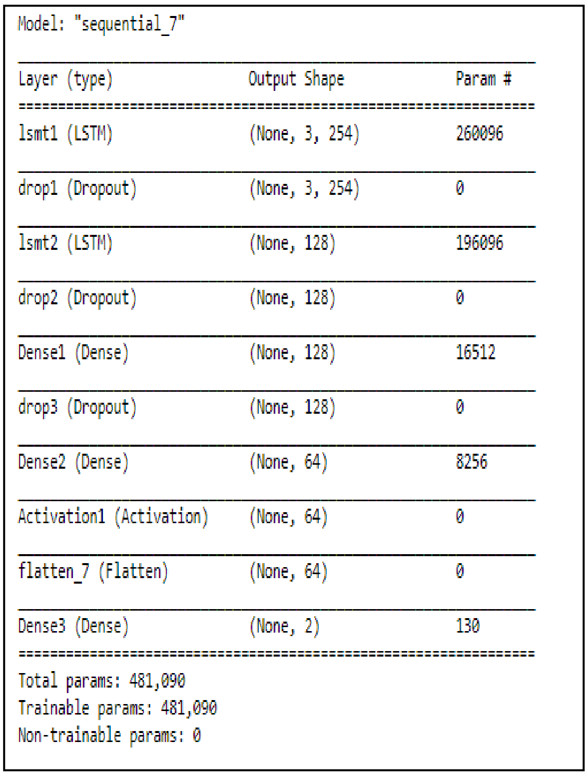
```
accuracy score:99.768519
recall score:99.122807
roc score:99.561404
             pred-flood  pred-none
true flood          318          0
false flood           1        113
```

Figure 21CNN evaluation

5.4.2 RNN

5.4.2.1 Architecture

RNN model is designed with Traditional neural network model structure that has a first layer LSTM block with 254 filters LSTM has a special architecture which enables it to forget the unnecessary information .,Next is Dropout layer operation on the input and return the output,LSTMnext layer ,Dropout ,Dense ,Dropout ,Dense ,activation function is tanh, flatten,In final the model, Dense layers. The developed RNN architecture is shown in Figure(22-23).



Model: "sequential_7"

Layer (type)	Output Shape	Param #
lsmt1 (LSTM)	(None, 3, 254)	260096
drop1 (Dropout)	(None, 3, 254)	0
lsmt2 (LSTM)	(None, 128)	196096
drop2 (Dropout)	(None, 128)	0
Dense1 (Dense)	(None, 128)	16512
drop3 (Dropout)	(None, 128)	0
Dense2 (Dense)	(None, 64)	8256
Activation1 (Activation)	(None, 64)	0
flatten_7 (Flatten)	(None, 64)	0
Dense3 (Dense)	(None, 2)	130

Total params: 481,090
Trainable params: 481,090
Non-trainable params: 0

Figure 22Details of RNN architecture and result output shape each layer

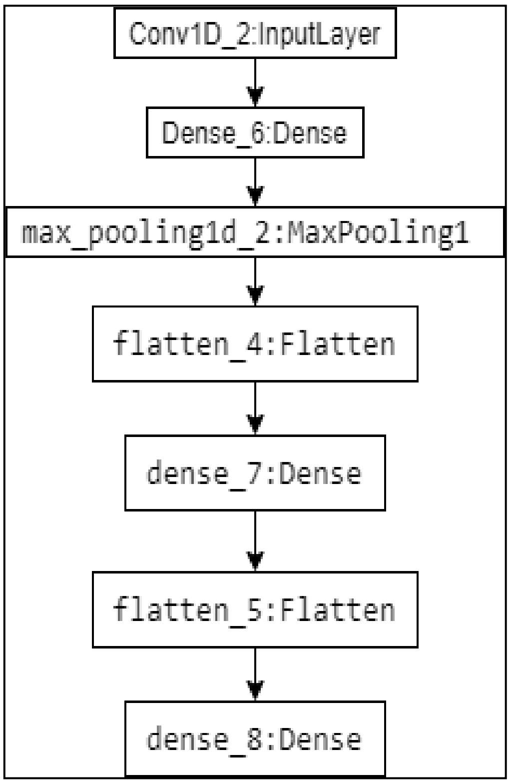


Figure 23Model architecture

5.4.2.2 Model Improvement and Training Procedure

We see the accuracy in the training and testing process in Figure 24, 25shows a representation of the loss in this process, it takes a few hours to complete this process.Here it turns out that when the number of epochs is 50, batch size is 5, we will obtain the highest verification accuracy (0.99%) corresponding to the lowest validation loss, so it is considered the optimal number of epochs.

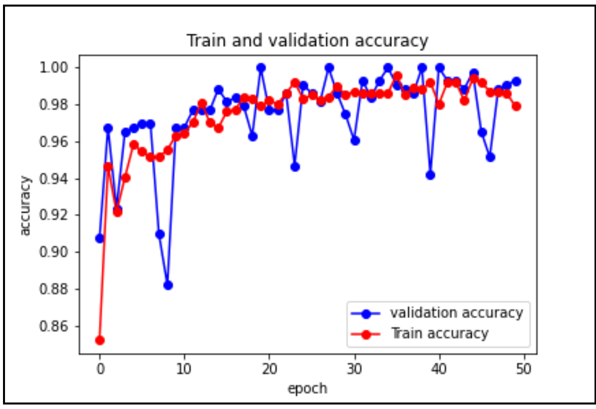


Figure 24accuracy of Training and validation process

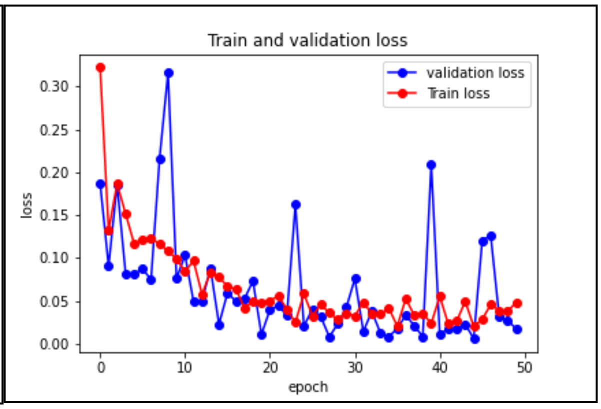


Figure 25loss of Training and validation process

```
Epoch 42/50
202/202 [==============================] - 2s 12ms/step - loss: 0.0282 - accuracy: 0.9899 - val_loss: 0.0172 - val_accuracy: 0.
9931
Epoch 43/50
202/202 [==============================] - 2s 12ms/step - loss: 0.0237 - accuracy: 0.9923 - val_loss: 0.0171 - val_accuracy: 0.
9931
Epoch 44/50
202/202 [==============================] - 3s 13ms/step - loss: 0.0368 - accuracy: 0.9868 - val_loss: 0.0230 - val_accuracy: 0.
9884
Epoch 45/50
202/202 [==============================] - 3s 16ms/step - loss: 0.0181 - accuracy: 0.9937 - val_loss: 0.0062 - val_accuracy: 0.
9977
Epoch 46/50
202/202 [==============================] - 3s 14ms/step - loss: 0.0248 - accuracy: 0.9944 - val_loss: 0.1198 - val_accuracy: 0.
9653
Epoch 47/50
202/202 [==============================] - 2s 12ms/step - loss: 0.0961 - accuracy: 0.9781 - val_loss: 0.1269 - val_accuracy: 0.
9514
Epoch 48/50
202/202 [==============================] - 2s 12ms/step - loss: 0.0670 - accuracy: 0.9808 - val_loss: 0.0322 - val_accuracy: 0.
9884
Epoch 49/50
202/202 [==============================] - 2s 12ms/step - loss: 0.0234 - accuracy: 0.9919 - val_loss: 0.0272 - val_accuracy: 0.
9907
Epoch 50/50
202/202 [==============================] - 2s 12ms/step - loss: 0.0353 - accuracy: 0.9855 - val_loss: 0.0177 - val_accuracy: 0.
9931
32/32 [==============================] - 0s 7ms/step - loss: 0.0219 - accuracy: 0.9921
Train Loss: 0.021908000111579895  Train accuracy: 0.9920634627342224
14/14 [==============================] - 0s 6ms/step - loss: 0.0177 - accuracy: 0.9931
Test Loss: 0.017738137394189835  Test accuracy: 0.9930555820465088
```

Figure 26RNN Training process and testing process results

When looking at the Figure 26 that represents the results of training the model and figure 27 below shows the correct cases for the prediction.

```
accuracy score:99.305556
recall score:97.478992
roc score:98.739496
             pred-flood  pred-none
true flood          313          0
false flood           3        116
```

Figure 27RNN evaluation

5.4.3 Multilayer perceptron (MLP)

5.4.3.1 Architecture

Where talk about building MLP we design two model with change value of parameters of Random Forest such as**alpha** L2 penalty (regularization term) parameter,**hidden_layer_sizes**The ith element represents the number of neurons in the ith hidden layer, **max_iter** Maximum number of iterations.First model assign value to max_iter=200,alpha=0.001,random_state=0,hidden_layer_sizes=1000 then we got the result of Accuracy on training set: 0.96, Accuracy on test set: 0.95.Second model assign value to max_iter=2000, alpha=0.1, random_state=0, hidden_layer_sizes=1000 then we got the result of Accuracy on training set: 0.993, Accuracy on test set: 0.979.Third model assign value to max_iter=200, alpha=0.00001, random_state=0, hidden_layer_sizes=3000 then we got the result of Accuracy on training set: 0.0.0.998, Accuracy on test set: 0.0.981.When we look at the results we obtained of prediction, it becomes clear to us that the second model is the best, and the figure 28 show the details

```
accuracy score:99.305556
recall score:97.478992
roc score:98.739496
             pred-flood  pred-none
true flood          313          0
false flood           3        116
```

Figure 28MLP evaluation

5.4.4 Computing environmentof deep learning algorithms

Typical training in this project was conducted on Windows 10 PC equipped with Intel core i7 CPU, NVIDIA 2GB GeForce 820M GPU. The MLP model was developed using scikit-learn 0.23 and CNN, RNN models was developed using TensorFlow 2.4.1 and Keras 2.4.3.

5.5 Models evaluation

After the process of building the models and training them on the training and testing data, now we compare the results between the models in the table below to find out the best models.

Table 1: models evaluation comparison

	Algorithm	Accuracy	recall	precision	F1
Machine Learning Algorithm	k-Nearest Neighbours	99%	100%	99%	99.4%
	Logistic regression	84%	86%	92%	87%
	Decision Tree	98%	98%	99%	98.4%
	Random Forest	97%	100%	99%	99.4%
	Support Vector Machine	98%	97%	98%	97%
Deep Learning Algorithm	CNN	99%	99%	99%	99%
	RNN	99%	97%	98%	97%
	Multilayer perceptron	94%	87%	92%	44.7%

By looking at the results above, we notice the superiority of Random Forest an algorithm in machine learning and CNN an algorithm in deep learning, where the two algorithms had the same results, both of them in classification have only one wrong classification.

5.6 Examples of Predictions of best model

We can use our ergonomically designed models CNN, KNN to make predictions about flooding. Table2 show examples of flooding projections.

Table 2 Example of Predictions

algorithm	data	prediction
KNN	[30.1,17.0,3]	flooding
CNN	[22.8,5.5,1]	Non-flooding

6. Conclusions and Future Studies

In this paper, an approach to flood forecasting is proposed. The system combines machine learning and deep learning to make it more efficient and convenient. During this project, more than one model was developed and results were compared between them to classify weather data and forecast floods. Models were trained with highly balanced data sets. Models (i.e. binary classification) were trained to classify data in two conditions: flood, non-flood. Model attained at accuracy of 99.768%. However, the measurement performance under balanced data conditions is adequate. This project will be expanded by exploring ways to increase the accuracy of the multi-category classification models. An immediate extension of this project is to check the performance of the model after adding additional blocks/layers to the existing CNN, RNN model and modifying hyperparameters, adding change to the machine learning models to achieve high accuracy that competes with current models. The model developed by CNN, KNN will also be improved by integrating with IOT system to work with sensor to work with data real time.

References

[1] E. Danso-Amoako, M. Scholz, N. Kalimeris, Q. Yang, and J. Shao, "Predicting dam failure risk for sustainable flood retention basins: A generic case study for the wider Greater Manchester area," *Comput. Environ. Urban Syst.*, vol. 36, no. 5, pp. 423–433, 2012.

[2] K. Xie, K. Ozbay, Y. Zhu, and H. Yang, "Evacuation zone modeling under climate change: A data-driven method," *J. Infrastruct. Syst.*, vol. 23, no. 4, p. 4017013, 2017.

[3] A. K. Lohani, N. K. Goel, and K. K. S. Bhatia, "Improving real time flood forecasting using

fuzzy inference system," *J. Hydrol.*, vol. 509, pp. 25–41, 2014.
[4] G. Corani and G. Guariso, "Coupling fuzzy modeling and neural networks for river flood prediction," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.*, vol. 35, no. 3, pp. 382–390, 2005.
[5] E. Todini, "Role and treatment of uncertainty in real-time flood forecasting," *Hydrol. Process.*, vol. 18, no. 14, pp. 2743–2746, 2004.
[6] T. Kerh and C. S. Lee, "Neural networks forecasting of flood discharge at an unmeasured station using river upstream information," *Adv. Eng. Softw.*, vol. 37, no. 8, pp. 533–543, 2006.
[7] M. Othman, A. A. Latif, S. S. Maidin, M. F. M. Saad, and M. N. Ahmad, "Engagement of Local Heroes in Managing Flood Disaster: Lessons Learnt from the 2014 Flood of Kemaman, Terengganu, Malaysia," *Achiev. Challenges Integr. River Basin Manag.*, p. 85, 2018.
[8] B. Yadav and K. Eliza, "A hybrid wavelet-support vector machine model for prediction of lake water level fluctuations using hydro-meteorological data," *Measurement*, vol. 103, pp. 294–301, 2017.
[9] N. Razali, S. Ismail, and A. Mustapha, "Machine learning approach for flood risks prediction," *IAES Int. J. Artif. Intell.*, vol. 9, no. 1, pp. 73–80, 2020, doi: 10.11591/ijai.v9.i1.pp73-80.
[10] N. Zehra, "Prediction Analysis of Floods Using Machine Learning Algorithms (NARX & SVM)."
[11] S. Sankaranarayanan, M. Prabhakar, S. Satish, P. Jain, A. Ramprasad, and A. Krishnan, "Flood prediction based on weather parameters using deep learning," *J. Water Clim. Chang.*, vol. 11, no. 4, pp. 1766–1783, 2020, doi: 10.2166/wcc.2019.321.
[12] D. N. Kumar, K. S. Raju, and T. Sathish, "River flow forecasting using recurrent neural networks," *Water Resour. Manag.*, vol. 18, no. 2, pp. 143–161, 2004.
[13] F. Y. Dtissibe, A. A. A. Ari, C. Titouna, O. Thiare, and A. M. Gueroui, "Flood forecasting based on an artificial neural network scheme," *Nat. Hazards*, vol. 104, no. 2, pp. 1211–1237, 2020.
[14] K. Kunverji, K. Shah, and N. Shah, "A Flood Prediction System Developed Using Various Machine Learning Algorithms," *Available SSRN 3866524*, 2021.
[15] Rami Ali, "Predictive Modeling," 2020. https://www.netsuite.com/portal/resource/articles/financial-management/predictive-modeling.shtml (accessed Aug. 05, 2020).
[16] I. El Naqa and M. J. Murphy, "What is machine learning?," in *machine learning in radiation oncology*, Springer, 2015, pp. 3–11.
[17] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.