IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 28, 2020, accepted April 2, 2020, date of publication April 6, 2020, date of current version April 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2986090

IoT-Enabled Flood Severity Prediction via Ensemble Machine Learning Models

MOHAMMED KHALAF[1], HAYA ALASKAR[2], ABIR JAAFAR HUSSAIN[3], (Member, IEEE), THAR BAKER[3], (Member, IEEE), ZAKARIA MAAMAR[4], (Member, IEEE), RAJKUMAR BUYYA[5], (Fellow, IEEE), PANOS LIATSIS[6], (Senior Member, IEEE), WASIQ KHAN[3], (Member, IEEE), HISSAM TAWFIK[7], (Member, IEEE), AND DHIYA AL-JUMEILY[3], (Senior Member, IEEE)

[1]Department of Computer Science, Al Maarif University College, Rumadi 31001, Iraq
[2]Computer Science, Prince Salman University, Riyadh 11586, Saudi Arabia
[3]Faculty of Engineering and Technology, Liverpool John Moores University, Liverpool L3 3AF, U.K.
[4]College of Information Systems, Zayed University, Dubai, United Arab Emirates
[5]CSSE, The University of Melbourne, Melbourne, VIC 3010, Australia
[6]Department of Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates
[7]School of Built Environment, Engineering and Computing, Leeds Beckett University, Leeds LS1 3HE, U.K.

Corresponding author: Thar Baker (t.baker@ljmu.ac.uk)

ABSTRACT River flooding is a natural phenomenon that can have a devastating effect on human life and economic losses. There have been various approaches in studying river flooding; however, insufficient understanding and limited knowledge about flooding conditions hinder the development of prevention and control measures for this natural phenomenon. This paper entails a new approach for the prediction of water level in association with flood severity using the ensemble model. Our approach leverages the latest developments in the Internet of Things (IoT) and machine learning for the automated analysis of flood data that might be useful to prevent natural disasters. Research outcomes indicate that ensemble learning provides a more reliable tool to predict flood severity levels. The experimental results indicate that the ensemble learning using the Long-Short Term memory model and random forest outperformed individual models with a sensitivity, specificity and accuracy of 71.4%, 85.9%, 81.13%, respectively.

INDEX TERMS Internet of Things, ensemble machine learning, flood sensor data, long-short term memory.

I. INTRODUCTION

Flooding is a natural disastrous phenomenon, having devastating impacts on goods, services, properties, and animal/human lives. An early warning about such a disaster might be helpful to mitigate the consequences and save lives. Though river flooding cannot be avoided, its impact may be minimized and controlled through appropriate planning and adoption of technologies such as IoT and ML.

According to a recent research study by Gartner [1], 6.4 billion connected objects/things were identified in 2016, representing an increase of over 3% compared to 2015, and expected to reach 20.8 billion by 2020. Some of these 'things' include a variety of sensors that might be useful for improving the quality of data collected for the purpose of making better decisions. IoT is an increasingly growing topic and widely available for such purposes [2], [3]. It permits things to be controlled or sensed remotely across several network environments, providing an interface for direct control over the physical world [4]. To extract useful and effective data, ML offers an appealing method for predicting water levels, for example. The vast majority of environmental monitoring centers have adopted IoT to assist in environmental protection [5], [6].

Sun and Scanlon [7] indicate that the use of machine learning has significantly improved the detection of early flood warning using powerful deep learning algorithms. Panchal *et al.* [8] show that gait characteristics can be utilised to capture flood levels and used machine learning algorithms including support vector machine and random forest for the analysis of the data. While Furquim *et al.* [9] propose a flood detection system based on IoT, machine learning and Wireless Sensor Networks (WSNs) in which fault-tolerance

The associate editor coordinating the review of this manuscript and approving it for publication was Moayad Aloqaily.

was embedded in their system to anticipate any risk of communication breakdown. Belal *et al.* [10] indicate that the lack of information about the quality of drinking water and the difficulty of early prediction of the flood has inspired various researchers to monitor and detect flood. The authors highlighted the importance of gathering reliable and quality data for the validity of the analysis.

In this research, we use multi-sensor data that originate from monitoring flood centres located in different countries around the world to determine rivers' water levels. To this end, a variety of advanced predictive models and learning algorithms were developed (i.e., Artificial Neural Networks (ANN), Random Forest (RF), K-Nearest Neighbour classifier (KNN), Long-Short Term Memory (LSTM) and Support Vector Machine (SVM)). The aim is to utilize machine learning algorithms to analyse flood sensors log datasets, characterized by nonlinearities and dynamic characteristics.

Despite the abundance of raw data, quality remains a concern to decision makers. For instance, missing values, get corrupted during transfer, and/or affected by noise are the most common factors that affect the data quality. To address this, a data science approach is adopted in this research for the analysis and feature extraction of sensory datasets, characterized by class imbalance, noise and missing values. Various ML techniques are used to analyze the flood sensor data. In the first round of simulation experiments, the single ML classification models were used that did not provide satisfactory performance and accuracy. Then, a new classifier which is based on the ensemble learning, was developed using LSTM, ANN and RF. Statistical results indicate the superiority of ensemble learning model over the all single/individual ML classifiers.

Therefore, the ''main'' contributions of this paper are as follow:

- Ensemble various machine learning algorithm to predict the river levels severity using IoT sensor river data.
- Improving the process of multi-levels classification and accuracy of our classifiers using deep learning algorithm such as LSTM.

The remainder of this paper is structured as follows. IoT and its application in disaster management scenarios is discussed in Section 2. The classification of flood sensor data is discussed in Section 3 along with the algorithms used in this research work. Section 4 discusses the methodology. The experimental design is described in Section 5, while Section 6 presents the simulation results. This is followed by the discussion and analysis in Section 7. Finally, the conclusions and avenues for further research are presented in Section 8.

II. DISASTER MANAGEMENTS USING IoT

Since 1980s, the U.S. has sustained with over 200 weather and climate disasters, with cumulative costs exceeding \$1.1 trillion [11]. Despite restrictions when it comes to reduced size, restricted connectivity, continuous mobility, limited energy [12], and constrained storage, IoT has a role to play in avoiding and/or offsetting the consequences of disaster recovery [13], [14]. However, on the other side, the reduced size and continuous mobility of IoT devices [15] (e.g., micro cameras mounted on drones) provides them with a unique advantage for use in restricted access areas. Moreover, their limited energy and constrained storage features make disposable IoT devices ideal for one-time situation assessment. Indeed, some IoT devices, such as sensors, are major stakeholders in the design and implementation of any preventive strategy. Such devices can collect and transmit large amounts of data to modelling and decision support-based systems, thus supporting the risk mitigation and the proactive deployment of emergency teams. Many studies have demonstrated the benefits of developing disaster recovery strategies based on IoT [16]–[19]. In a disaster recovery situation, all 'things' including virtual or real entities such as human beings, inanimate objects, intelligent software agents and even virtual data could; contribute positively [20]. Maamar *et al.*, in [21] introduce the concept of Process-of-Things (PoT) to allow the collaboration of living and no-living things [22]. A PoT is specified as a story that ''tells'' how to discover and select things based on their capabilities, how to support things taking over new/adjusting their capabilities, how to facilitate the (dis)connection of things using pre-defined (social) relations, and how to incentivize/penalize things because of their constructive/destructive behaviors, respectively. Maamar *et al.*, exemplified PoT within a healthcare scenario [23], where things representing medical equipment (e.g., thermometers), ambient facilities (e.g., air sensors), patients (e.g., smart wrists), and care providers (e.g., doctors) work together. Such a scenario can be easily related to a disaster recovery situation. In another work [24], Soubhagyalaxmi *et al.*, discuss a disaster management system using IoT in India. The study indicates that around 57% of the land is vulnerable to earthquakes. Of these, 12% is vulnerable to severe earthquakes, 68% of the land is vulnerable to drought, 12% of the land is vulnerable to flood, 8% of the land is vulnerable to cyclones, and many cities in India are also affected by chemical, industrial and other types of man-made disasters. Likewise, Lempitsky *et al.*, [25] present the importance of IoT in natural risk management in terms of detection, prevention, and management with an economic evaluation of each stage.

III. CLASSIFICATION OF FLOOD SENSOR DATA

In recent years, a variety of ML techniques have been developed to handle the high dimensional datasets in diverse application domains [26]. These techniques can analyse large numbers of attributes and represent each object of interest with a distinct label. In short, a ML model learns to perform the mapping (or equivalently, approximate a function) between the input (feature) space and the output (object class) space. Each input vector corresponds to an object $\boldsymbol{x}$, characterized by a set of features, whereas $\boldsymbol{y}$ describes the class label assigned to $\boldsymbol{x}$. In this respect, the classification

TABLE 1. **ml classifiers parameters and description.**

Architecture	Training Algorithm	Parameters
ANN (Multilayers Perceptron)		
Inputs: 11 Hidden Layers: 2 Activation Functions: Tansig Outputs: 3 classes	Levenberg-Marquardt	Adaptive learning rate (LR) settings: initial value: 0.001 coefficient for increasing LR: 10 coefficient for decreasing LR: 0.1 maximum learning rate: $1e^{10}$
RF (Random Forest)		
Inputs: 11 Number of Trees: 300 Outputs: 3 classes	Random feature bagging	Total number of decision trees to be produced 200. Total number of feature subsets: 1
SVM (Support vector Machine)		
Inputs: 11 Outputs: 3	Quadratic Optimization	Kernel: Polynomial
KNN (k-nearest-neighbours classifier)		
Inputs: 11 Neighbours: 4 Classes: 3	K nearest neighbours measured by the distance function.	Euclidean distance
LSTM (Long-Short Term Memory)		
Layers: 2	error backpropagation algorithm	Learning rate: 0.0001 - Number of units: 20; 30; 50 Number of epochs: 100,000 with early stopping

process employed to label training and testing data sets is also known as a descriptive classifier, i.e., a method to discover the class label for various inputs [27], [28].

To apply classification in the target research domain, it is vital to identify distinctive feature patterns from un-labelled datasets during the testing process.

Equation (1) represents the class of linear classifiers [29].

$$g(x) = w^{\mathrm{T}}x + b \tag{1}$$

where $g(x)$ represents the linear function with input 'x', 'w' is the weight vector and 'b' is the bias term. For two classes, e.g., c_1 and c_2, the linear classifier results to a target class of c_1, if g(x)>0, and a class of c_2, when g(x)<0. However, since the real-world data is often non-linear in nature, a nonlinear classifier is needed to be used that can capture the nonlinearities in the data and provide high classification performance, which is vital when it comes to predicting the onset of natural disaster phenomena. For the classification and analysis of both linear and nonlinear flood-sensor data, various classification techniques are used that include RF, SVM, ANN, KNN, and LSTM. Table 1 shows the configurations of the classifiers used in this research.

A. RANDOM FOREST CLASSIFIER

Random Forest (RF) is utilised for regression and classification tasks. It is a series of decision trees, each of which acts as a weak classifier, typically characterized by poor prediction performance, however in aggregate form, it offers robust prediction. Therefore, this classifier can be thought of as a meta-learning model. RF was originally proposed by Ho [30], [31], and subsequently enhanced by Ho [32], with the latter being widely used in recent studies. RF uses feature bagging and decision trees structures [33].

RF efficiently and effectively produces partitions of high-dimensional features based on the divide-and-conquer strategy, over which a probability distribution is located. Moreover, it permits density estimation for arbitrary functions, which can be used in clustering, regression, and classification tasks. Classification results are obtained by averaging the decisions formed through the layers of the forest, permitting the collective knowledge of the decision-tree learners to be incorporated. Equations 2 and 3 summarize the RF.

$$f(x) = \frac{1}{m}\sum\nolimits_{i=1}^{m} f(x, x_i p) \tag{2}$$

where x is the partial dependence variable and $x_i p$ refers to the data variable

$$f(x) = \log t_j - \frac{1}{J}\sum\nolimits_{k=1}^{J} (\log t_k(\mathrm{y})) \tag{3}$$

where 'J' refers to the number of classes (3 in our case), and 'j' refers to the individual class (i.e. normal/abnormal/dangerous water levels in this study). In addition, t_k belongs to the proportion of total votes for class 'j'.

B. SUPPORT VECTOR MACHINES

Support Vector Machines (SVMs) is a type of supervised learning and can be used for classification and regression problems. SVM is based on soft margin classification [34], which lends itself on concepts of statistical method theory [35]. Given a training dataset containing instance-label pairs $(x_1, y_1), \ldots, (x_N, y_N)$ where $x_i \in R^d$ and $y_i \in \{-1, +1\}$, SVM solves the optimization problem:

$$\begin{aligned} &\min_{x,b,\xi_i} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \\ &\text{subject to } y_i\left(\langle \emptyset(x_i), w\rangle + b\right) \\ &\qquad -1 + \xi_i \geq 0 \\ &\qquad -\xi_i \geq 0, i = 1\ldots, N \end{aligned} \tag{4}$$

where $\emptyset(x_i)$ is a non-linear kernel that maps the training data onto a high-dimensional space. To separate the two classes, SVM works by finding the separating hyperplane that maximizes the margin between observations. The slack variables ξ_i allow misclassification of difficult or noisy patterns and $C > 0$ is the regularization parameter, which controls the degree of overfitting. Finding the support vectors is made possible using the Lagrange multipliers a_i allowing Equation 4 to be rewritten into its dual form:

$$\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha$$
$$\text{subject to } y^T\alpha = 0$$
$$0 \leq \alpha_i \leq C, \quad i = 1, \ldots, N \tag{5}$$

where $e = [1, \ldots 1]^T$ is a vector of ones, Q is a m-by-m matrix, $Q_{i,j} \equiv y_i y_j K(x_i, x_j)$, and $K(x_i, x_j) \equiv \emptyset(x_i)^T \emptyset(x_j)$ is the kernel function. Using this primal-dual relationship, the optimal w can be found:

$$w = \sum_{i=1}^{N} y_i \alpha_i \emptyset(x_i) \tag{6}$$

Then the new samples are classified by:

$$sgn\left(w^T \emptyset(x) + b\right) = sgn(\sum_{i-1}^{N} y_i \alpha_i K(x_i, x)) \tag{7}$$

As previously mentioned, the separating hyperplane is found by solving the optimization problem, which allows the selection of the support vectors that maximize the margin between the two classes (e.g. normal vs abnormal water level in this case). The penalty parameter C is a critical tuning parameter for constructing a good model that generalizes well. In addition, several kernel functions are available to support the transformation of the input data into a higher-dimensional space, where linear separability is possible. An example of such a function is the polynomial of order d:

$$K(x, y) = (1 + x \cdot y)^d \tag{8}$$

C. ARTIFICIAL NEURAL NETWORKS

Neural Networks is a problem-solving approach based on a connectionist model. It comprises networks of interconnected neurons, whose weights are adapted until a solution emerges. Artificial Neural networks are inspired by biological neural networks found in the mammalian brain, to design and mimic the information processing capability of such powerful biological structures. A Multilayer Perceptron (MLP) is a 'feedforward' neural network, where information is transferred forward. The input information is presented to the network in the input layer, which transfers the information into a sequence of one or more hidden layers before it is represented as a pattern in the output layer. The transfer functions of the neurons in the hidden layers, and typically the output layer, are nonlinear. During the training process, the backpropagation algorithm is used to compute and updates weights in response to the error feedback. MLPs have been successfully used in many applications, including signal processing [36], facial expression analysis [37] and function approximation [38].

Several studies have been conducted to speed up and enhance the accuracy and performance of backpropagation learning. For instance, the use of a momentum term was proposed in [39], weight decay was used in [40], and variable learning rate was reported in [41]. Alternative neural computing approaches attempted to mimic additional biological neuronal characteristics such as axonal delays for improving performance [42]. A method that combines the advantages of gradient descent and second-order information is the Levenberg-Marquardt algorithm [39], which is used in the experiments of this research.

D. K-NEAREST NEIGHBOUR CLASSIFIER

K-Nearest Neighbour Classifier (KNN) is a lazy supervised learning algorithm that has been applied in various fields, such as statistical analysis, data mining, and pattern recognition [43]. Classifying data is based on the closest training samples in feature space. The aim is to classify the new (unseen) input patterns, based on a nominated number of their nearest neighbours. KNN is appropriate when there is no prior knowledge about the distribution of the data [44]. It works well in both training and testing [45]. The method consists of four steps. Firstly, training examples with their labels are stored in feature space. Next, the number of 'k' nearest neighbours and the appropriate distance metric are selected. Unseen data is fed into KNN and their nearest 'k' instances are found. Each of the new data is assigned the label corresponding to the majority label of its neighbours. The detailed description and mathematical formulation of KNN classifier can be found in [46].

E. LONG-SHORT TERM MEMORY (LSTM)

LSTM was introduced by Hochreiter and Schmidhuber in 1997 [47]. To elucidate the problem of gradient fading, the authors referred to the concept of memory cells. LSTM indicates the addition of short-term memory (STM) between memory cells since traditional neural networks have long-term memory in the weights of the connections. Other adaptations including peephole connections [48] have been suggested. These loop closure connections enhance performance as they do not need to go through the activation function, thus reducing computational complexity. This typology has excessive diffusion and utilisation nowadays, where complex LSTM networks, consisting of multiple layers are utilised with reasonable computational costs.

IV. METHODOLOGY

Various ML techniques have been used to predict the severity of flood disasters based on sensor data [49]. In this work, we compare the performance of popular ML methods (described in Section 3) on IoT flood sensor data and propose an ensemble approach for a novel flood severity classification system. Since a flood can take place in any geographical area, the system uses local sensor data. The objective is to classify data collected from the flood sensors into three categories, namely, normal, abnormal, and high-risk leading to flooding.

The proposed system comprises of sequential components including; data collection from IoT sensors, pre-processing, feature-space data representation, classification model configuration and training using the processed data, and finally, models' evaluation using the unseen testing data set.

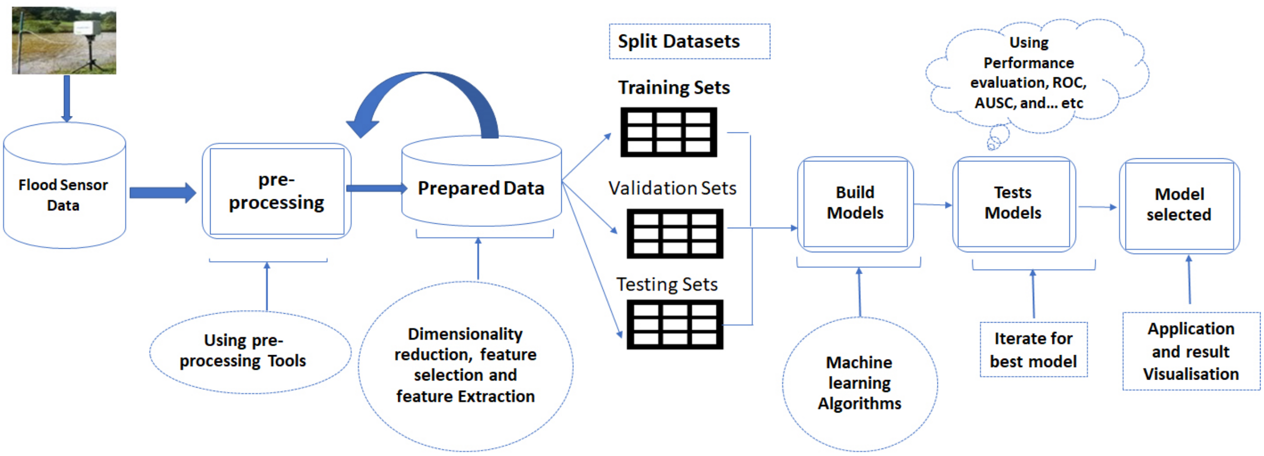

FIGURE 1. Overview of the data processing pipeline.

Figure 1 presents an overview of the sequential processing and analysis pipeline for the classification of flood sensor data. The processed flood sensor dataset is divided into three partitions for training, validation, and testing of the ML models. Eleven features (shown in Table 2) are fed into ML algorithms as inputs to perform classification of three output classes (see Table 2).

Our system consists from data collection stage in which flood sensory data information will be collected. The collected data will be forwarded to pre-processing stage, which will be performed for cleaning and normalization of the data. In this case, 11 attributes are collected including monsoonal rain, duration in day, number of dead, number of displaced, snowmelt and ice jams, magnitude, centroid, heavy rain, torrential rain, total of affected area and tropical storm. The cleaned and normalized data using the appropriate attributes are forwarded to trained machine learning algorithms are used to classify the severity of the flooding.

A. DATA COLLECTION

The sensory data is collected from the Environment Agency [50], that contains a collection of datasets from various cities around the world. Each sample of the flood sensor datasets contains 11 features deemed important for predicting the severity (as illustrated in Table 2). The flood sensor data used in this study consists of 4214 flood samples, with the three target classes defining the severity of the flood. The outputs were coded as class one with 1181 data points

(normal water level), class two with 306 data points (abnormal water level), class three 456 (high-risk water level). In terms of statistical procedures, the aggregate function is used to combine data from several measurements. Furthermore, when data is aggregated, groups of observations are substituted with summary statistics of the techniques, depending on those observations [51].

TABLE 2. Flood sensor dataset descriptions, selected algorithms and evaluation metrics.

No	Type	Number	Description
1	Data instances	4214	Data for a period of 31 years were collected from the Environment Agency website.
2	Class Variables	3	Class 1: Normal water level Class 2: Abnormal water level Class 3: Dangerous water level
3	Features (Attributes)	11	Monsoonal Rain, Duration in day, Number of Dead, Number of Displaced, Snowmelt, and Ice Jams, Magnitude, Centroid, Heavy Rain, Torrential Rain, Total of Affected area, Tropical Storm
4	Evaluation Metrics of classification models	5	F1 Score, Precision, Sensitivity, Accuracy, Specificity
5	Visualization Techniques	5	T-distributed Stochastic Neighbourhood Embedding (tSNE), Receiver operating characteristics (ROC) curve, Principal Component Analysis (PCA) and Area Under the Curve (AUC).
6	ML Algorithms	5	ANN, RF, KNN, LSTM and SVM
7	Ensemble Classifier	2	(ANN Combination, ANN models with RF) (LSTM Combination, LSTM model with RF and LSTM with SVM)

B. PRE-PROCESSING

To obtain accurate results with the use of ML techniques, it is important to appropriately prepare the aforementioned flood sensor data through pre-processing techniques, i.e., cleansing and normalization. Noise reduction and dealing with missing values are essential in ML and subsequently, for higher prediction accuracy and overall performance, which are performed following the data standardization tools explained in [52].

C. EXPLORATORY ANALYSIS

Exploratory analysis was utilised to identify possible outliers in the pre-processed data. This is an important step, as it

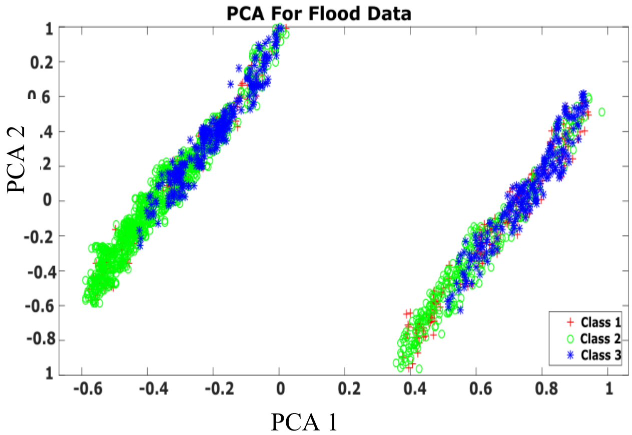

FIGURE 2. A projection of the three classes in the space of the first two Principal Components.

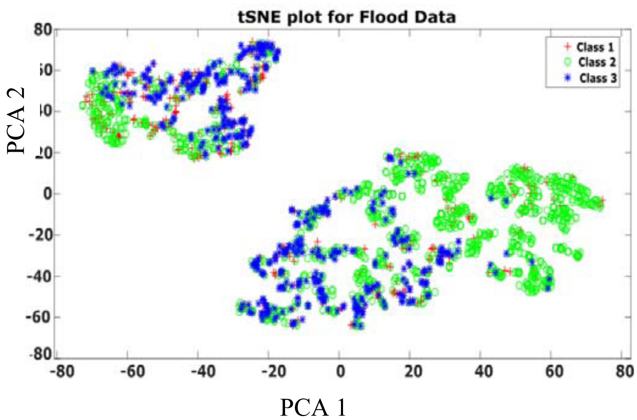

FIGURE 3. T-distributed Stochastic Neighbourhood Embedding Plot.

provides further insight and increase the efficiency in terms of learnability of the data set by training models. To illustrate this, a visualisation of the flood sensor dataset using PCA is depicted in Figure 2. PCA is commonly used for dimensionality reduction and finds a great deal of use in a variety of applications [53]. It performs a linear mapping so that potential correlations in feature space are minimized and could be used for projection into a lower dimensional space.

In Figure 4, a visualization of the three classes is shown in the space of the two principal components, i.e., the components which contain the highest amount of information, in terms of variability (variance) in the data. By evaluating the PCA plot, it is evident that the flood sensor data is not linearly separable.

Another type of visualisation we implemented is the T-distributed Stochastic Neighbourhood Embedding (tSNE) [54], as shown in Figure 3. This is in fact a machine learning algorithm, which, unlike PCA, results to a non-linear projection of high-dimensional data into a lower dimensionality, typically two or three dimensions, subspace. In t-SNE, patterns which share similarities in the original feature space are projected in close proximity, while dissimilar patterns are projected at distant points with high probability. By examining Figure 5, there are considerable similarities between patterns belonging to different classes, as the corresponding clusters in the projection space are overlapping.

D. FEATURE SELECTION

Feature selection is an important step for the analysis of standardized data, to enable our ML algorithms to train faster and overcome the problem of overfitting. There have been various approaches for the feature selection used in ML related problems including principal component analysis [42], information gain [55], chi-square test [56], [57] and many more. One of the key motivations of utilizing chi-square test in our approach is the way it ranks features based on statistical significance indicating the dependency between the current feature and the target class [57].

Let us consider the dimensionality of the original space to be ‘d’, while the dimensionality of the reduced feature space to be ‘r’, where $r < d$. In this case, ‘r’ is determined using the chi-square filtering approach to evaluate high-ranking features. If the chi-square test value is lower than a critical value (in our case we selected a value of 0.05) then the null hypothesis is selected and the feature is regarded as important; else, the null hypothesis is disregarded, and the feature is rejected. Algorithm 1 shows the proposed feature selection algorithm. Our algorithm selected 6 features which are considered important including magnitude, torrential rain, heavy rain, snowmelt and ice jams, monsoonal rain and tropical storm.

Algorithm 1 Feature Selection Sets

Let X be a set of data in form of signals such that $X = \{x| x \in S$ where S belongs to flood sensor data$\}$ & X^d where d is the dimensionality of the original feature space.

Let $X_c \in X$ where $X_c = \{x| x = \text{Filtered}(x)\}$ & X_c^r where r is the dimensionality of the selected feature space: r<d. Perform the following steps to identify the important feature set X_c:

Foreach feature *x* in **X** Do:
  Perform Chi-square test for current *x*
  IF Chi-square(x) < critical value (0.05 in this case)
    Accept *null* hypothesis
    Add *x* to X_c i.e. $x \in X_c$
  ELSE
    Accept alternative hypothesis (i.e., $x \notin Xc$)
  End IF
End Loop

V. EXPERIMENTAL DESIGN

The evaluation metrics used in the experiments measure the results of the ML techniques in flood prediction (refer to Table 3). The holdout approach is used to assess the generalization performance on an independent flood sensor dataset.

The main objective of this work is to predict floor severity levels using multi-sensor flood data and advanced machine learning algorithms. This can provide improved prediction

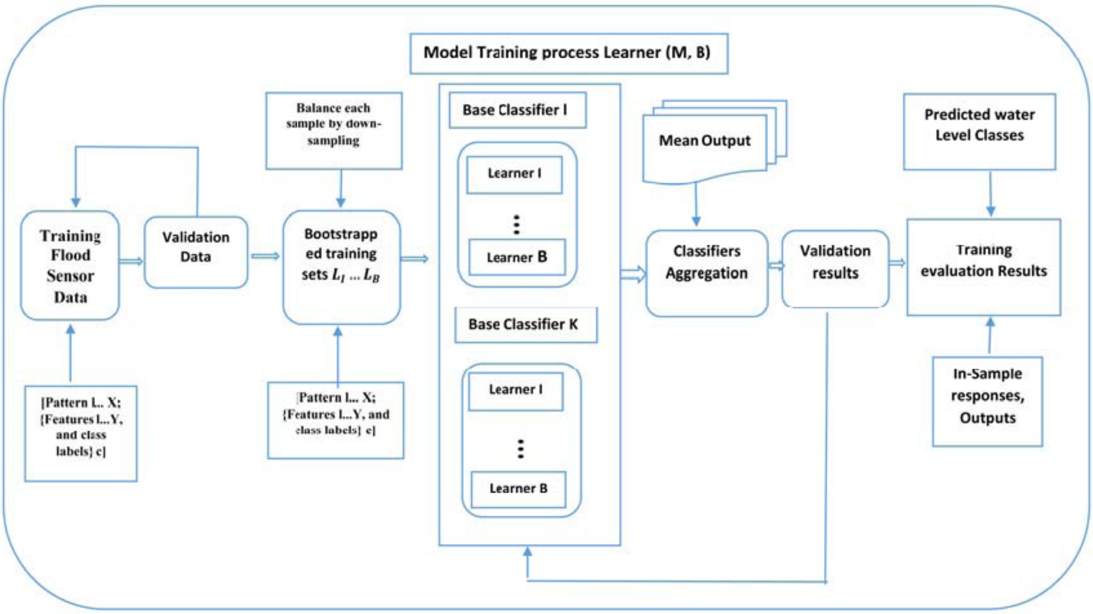

FIGURE 4. Ensemble Classifier (Training Process).

accuracy using the composition of data analysis and ML techniques to investigate the effect of integrating strong and weak classifiers and compare their performance with those of the individual classifiers.

Experiments (A, B) are designed using cross-validation (with 70% of the data used for the training, 10% for the validity and 20% for testing) to evaluate the performance of the proposed approach:

A) Multiple ML models (ANN, RF, KNN and LSTM) were trained and tested individually over the flood dataset.

B) Ensemble model is trained and tested over the flood dataset (LSTM+SVM, LSTM+RF, ANN +RF) while using voting and stacking to measure the performance.

Various statistical performance measures are utilised to benchmark our results including sensitivity, specificity and accuracy.

Algorithm 2 shows the proposed system.

A. ENSEMBLE CLASSIFIERS

Research studies indicated better classification performance obtained using a composition of multiple classifiers [58]. Figure 2 illustrates the block diagram for the proposed system. There are N input and output sets, $1, .., X$and$1, \ldots, Y$, respectively. The bootstrapping begins with a few training samples of $1, \ldots, x$ and $1, \ldots, y$ using the primitive model pool $1, \ldots B$. Base models are represented through $1, \ldots, M$. To estimate the accuracy and performance of the final output, we averaged the outputs of the base classifiers using predicted class probabilities. As in the case of single classifiers, classification performance is measured in terms of Sensitivity, Specificity, Precision, Accuracy, and F1 Score.

Algorithm 2 Proposed Algorithm for Sensor Data Classification and ML Selection

---

Let W, M, D, Dd, DI, TO, H, TS, SM, MS be various sets of values $\subset X_c$ determined as Algorithm 1

Let t, a and c are data values determined as Algorithm 1.

$n = 0$

Training $= \{t \in X_c\}$

Test $= \{ts \in X_c \;\&\; t \notin \text{Training}\}$

1: For every selected ML algorithm determined

2: E[Sensitivity] $= \{S: S \Rightarrow$ ML (Training, Test)$\}$

3: Calculate the average output and classifier aggregation

4: $n = n + 1$ if $n <$ Threshold, go to 1 else classify

5: $\forall$ ml $\in$ ML, if E[Sensitivity]$_{ml}$ $<$ Th, $\exists$ e $\in$ Ensemble| Ensemble $= \{x|$ x ensemble of various ML$\}$ & E[Sensitivity]$_e$ $>$ Th where Th is a threshold value representing the required classification accuracy.

For the ensemble classifier, we used the stacked and the voting methods. The stacked method concerned with combining several classifiers with various learning algorithms. In our experiment, each selected model is trained using the same training sets as is illustrated in Figure 4.

VI. CLASSIFICATION RESULTS

In this section, we present the results of validation and testing for the single and ensemble classifiers. In summary, it is shown in Table 4 that ensemble classifiers outperform single classifiers. The top-performing single classifier in terms of the sensitivity for the classification of high-risk flooding class was LSTM, which produced a value of 0.925 in training/validation, however, its generalization was not as good,

TABLE 3. Performance evaluation per Model (Validation/Training).

Mod	Target Class [C1, C2, C3]	SEN	Spec	Prec	F1	Accu
ANN	C 1	0.466	0.63	0.644	0.541	0.533
	C 2	0.541	0.559	0.278	0.367	0.555
	C 3	0.525	0.597	0.213	0.303	0.584
RF	C 1	0.695	0.885	0.896	0.783	0.773
	C 2	0.85	0.797	0.567	0.68	0.809
	C 3	0.742	0.859	0.522	0.613	0.839
KNN	C 1	0.809	0.677	0.782	0.795	0.755
	C 2	0.817	0.776	0.533	0.645	0.786
	C 3	0.738	0.851	0.507	0.601	0.832
SVM	C 1	0.0035	0.9944	0.25	0.0070	0.656
	C 2	0.9310	0.0410	0.31154	0.4668	0.3239
	C 3	0.057	0.9611	0.4324	0.10	0.6528
LSTM	C 1	0.996	0.9574	0.9238	0.95876	0.9707
	C 2	0.781	0.9607	0.90265	0.83778	0.9037
	C 3	0.925	0.9371	0.88396	0.90401	0.9330
LSTM + RF	C 1	1	0.9963	0.9929	0.99644	0.9975
	C 2	0.98467	1	1	0.99228	0.9951
	C 3	1	0.996	0.9929	0.99644	0.9975
LSTM + SVM	C 1	0.99286	0.957	0.92359	0.95697	0.9695
	C 2	0.71264	0.975	0.93	0.80694	0.8915
	C 3	0.95714	0.903	0.8375	0.89333	0.9220
ANN + RF	C 1	0.985	0.913	0.942	0.963	0.956
	C 2	0.973	0.953	0.866	0.917	0.958
	C3	0.996	0.944	0.786	0.879	0.953

TABLE 4. Performance evaluation per Model (testing).

Mod	Target Class [C1, C2, C3]	SEN	Spec	Prec	F1	Accu
ANN	C 1	0.535	0.6	0.711	0.611	0.558
	C 2	0.529	0.534	0.236	0.326	0.533
	C 3	0.691	0.548	0.197	0.306	0.568
RF	C 1	0.795	0.507	0.748	0.771	0.693
	C 2	0.682	0.668	0.358	0.47	0.671
	C 3	0.564	0.501	0.153	0.241	0.51
KNN	C 1	0.434	0.693	0.723	0.542	0.525
	C 2	0.682	0.447	0.251	0.367	0.497
	C 3	0.436	0.536	0.131	0.202	0.523
SVM	C 1	0	0.9925	0	0	0.6536
	C 2	0.907	0.05	0.3072	0.4591	0.3219
	C 3	0.042	0.9333	0.25	0.07317	0.6292
LSTM	C 1	0.914	0.88148	0.8	0.8533	0.892
	C 2	0.538	0.81429	0.5737	0.5555	0.7268
	C 3	0.700	0.88889	0.7656	0.7314	0.824
LSTM + RF	C 1	0.885	0.9111	0.8378	0.86111	0.9024
	C 2	0.6	0.7714	0.5493	0.57353	0.717
	C 3	0.657	0.8963	0.76667	0.70769	0.8146
LSTM + SVM	C 1	0.9	0.88148	0.79747	0.84564	0.8878
	C 2	0.47	0.84286	0.58491	0.52542	0.7268
	C 3	0.7	0.85926	0.73973	0.75524	0.8292
ANN + RF	C 1	0.705	0.743	0.835	0.765	0.719
	C 2	0.706	0.716	0.403	0.513	0.714
	C 3	0.582	0.81	0.33	0.421	0.779

resulting to a sensitivity of 0.7 during testing. The worst performing classifier in terms of sensitivity was the SVM with the poor performance of 0.057 and 0.042 for the training and the testing, respectively. This indicated that selecting appropriate kernels for our three classes flood data may not be possible.

In our experiments, single classifiers produced ACC values of 0.557, 0.807, 0.867 and 0.935 for ANN, RF, KNN, LSTM respectively, during the validation/training. During testing, the performances of the classifiers are 0.582, 0.65, 0.538 and 0.81 for ANN, RF, KNN, and LSTM respectively. Ensemble classifiers demonstrate better validation performance, as shown in Table 4 and Figures 6-8 which relate to the ROC and AUC graphs, respectively. The combination of ANN and RF classifiers shows an accuracy of 0.956 in validation, however, this reduces to 0.737 in testing in terms of the average for the three classes (stated in Table 2) for ensemble classifiers. Improved results are obtained with the combination of LSTM and RF, which demonstrates a validation accuracy of 0.997 during training (Table 3), while in testing, this reduces to 0.811 (Table 4). Still, the strong generalization of the ensemble classifier confirms that there is valuable information within the flood sensor data that can be captured with such types of ML structures. Furthermore, the ensemble classifier for LSTM and RF showed an average sensitivity of 0.714, followed by the ensemble of the LSTM and SVM with an average sensitivity of 0.69, then the ensemble of ANN and RF with an average sensitivity of 0.664 for the testing data.

As single SVM classifier did not perform well during the training and the testing, the results for the ensemble classifier with SVM and LSTM indicate that the performance is based on the classification power of the LSTM rather than the SVM.

Tables 3, 4 and Figures 5, 6 demonstrate the results for each classifier for estimating the performance evaluation techniques of the models. We used the holdout methods dividing the flood data into the training phase, validation phase and testing phase. The training data represents 70%, while the validation data sets represent 10%, and testing sets represents 20%. To train the sensor data, it is crucial to perform two procedures: initially, we construct the first structure for each model based on the training set, to assess the error rates as shown in Figure 5, then based on the performance and accuracy that models received during the training

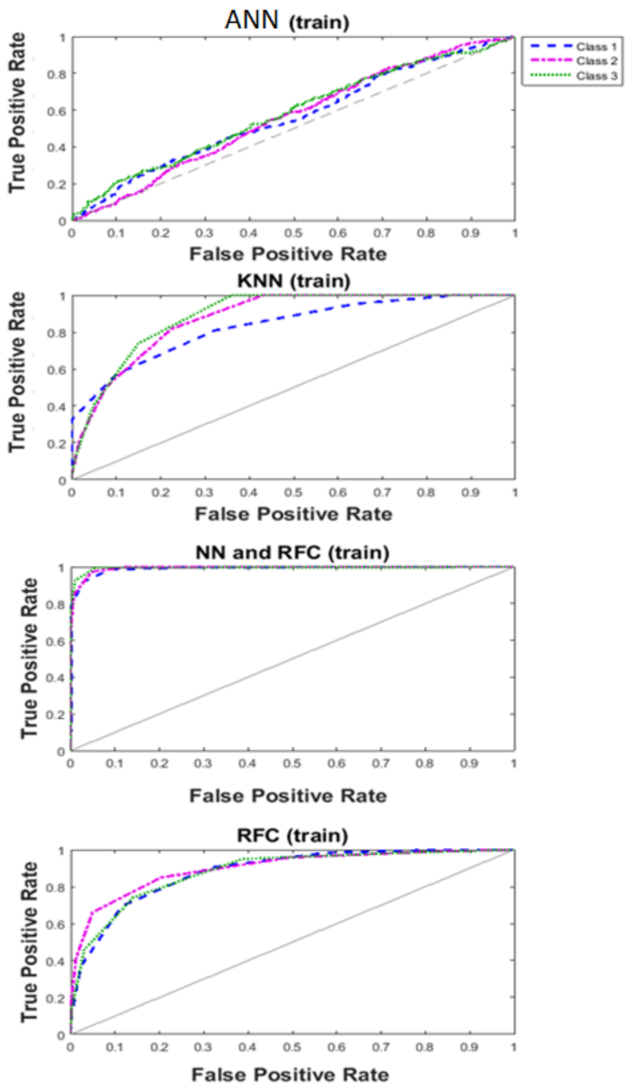

FIGURE 5. ROC Curve Per Classifier for Flood Sensor Data (Validation/Training Phase).

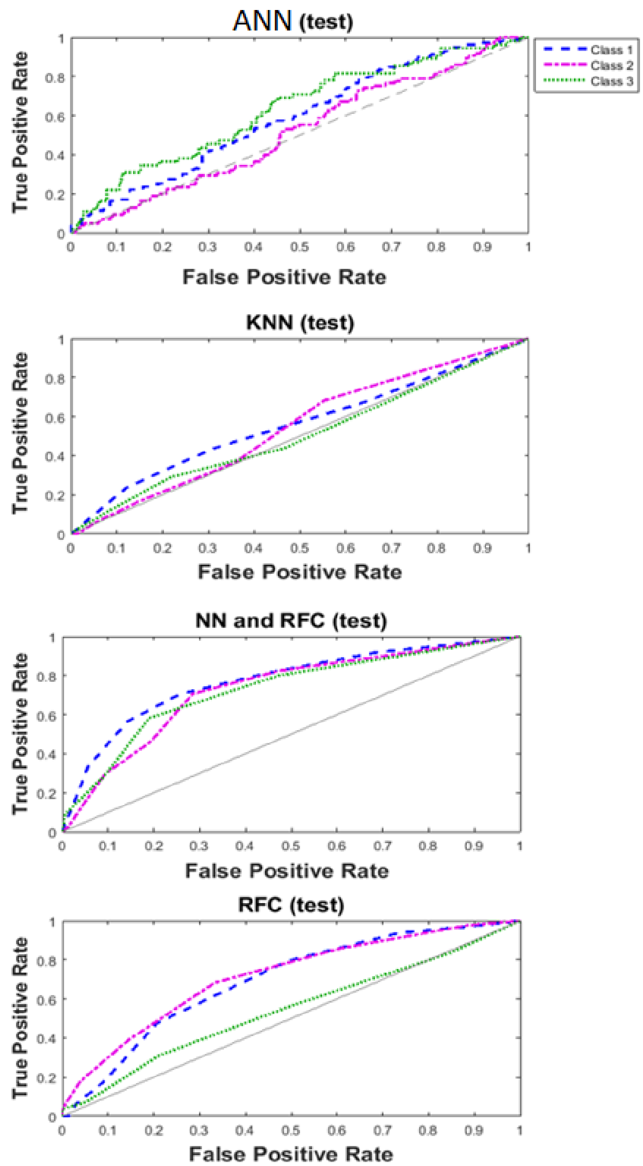

FIGURE 6. ROC Curve Per Classifier for Flood Sensor Data (Testing Phase).

sets, we estimate the error rate for each model as shown in Figure 6.

Figures 7 and 8 show the results for each class over the single classifiers and ensemble classifier. AUC shows outcomes for each model with three output targets. In AUC plots, the Y-axis demonstrates the corresponds to each model entries, whereas the X-axis shows the classes and classifier. When AUC yields one that refers to an ideal approach, while an AUC with 0.5 illustrates random generalization. Each bar in AUC plot is associated with a corresponding curve. The main advantage of using AUC figure is to offer a standard graphical form.

Compared to individual ML techniques, ensemble classifier indicated to acquire a great comparable performance and accuracy, besides being faster. In this aspect, the classification results in association with evaluation metrics based on confusion the table for three classes [class one, class two, and class three] are demonstrated in Tables 4 and 5. To elaborate that more, Ensemble classifier illustrated vital overlaps between inputs features and output classes. The ensemble classifier outperformed individual classifiers and yield such an acceptable performance and accuracy for three target output (classes) as shown in the ROC and AUC graphics. Furthermore, this also indicated that the ensemble classifier can handle multi-class problem compared to individual models that can handle only two classes problem.

The reason ensemble classifier outperforms individual models is due to the data variables' distributions. We found that flood sensor data has strong non-linearities within the variables. To build such as accurate classifier, it is significant to alter the parameters in each classifier. Accordingly, to handle large instances in our flood sensor data, neural

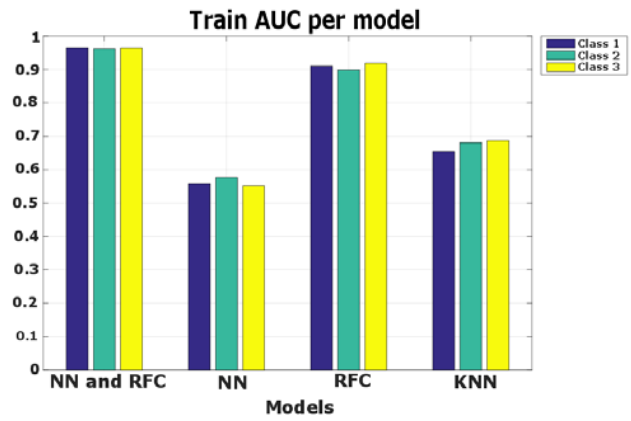

FIGURE 7. AUC per Classifier for Flood sensor data (Training Process).

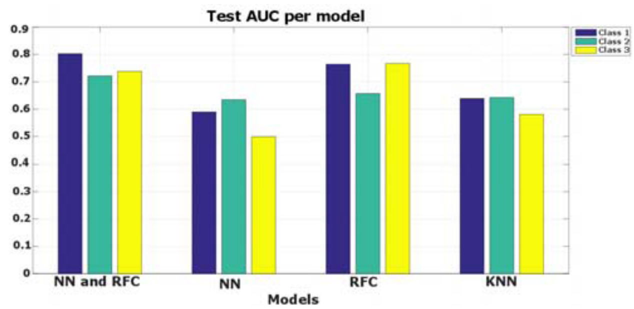

FIGURE 8. AUC per Classifier for Flood sensor data (Testing Process).

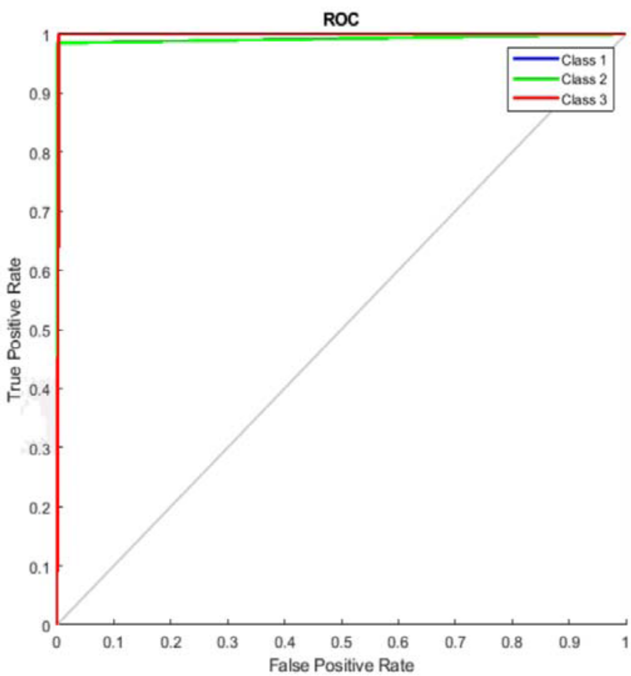

FIGURE 9. ROC for the training data for flood sensor using ensemble (LSTM + RF).

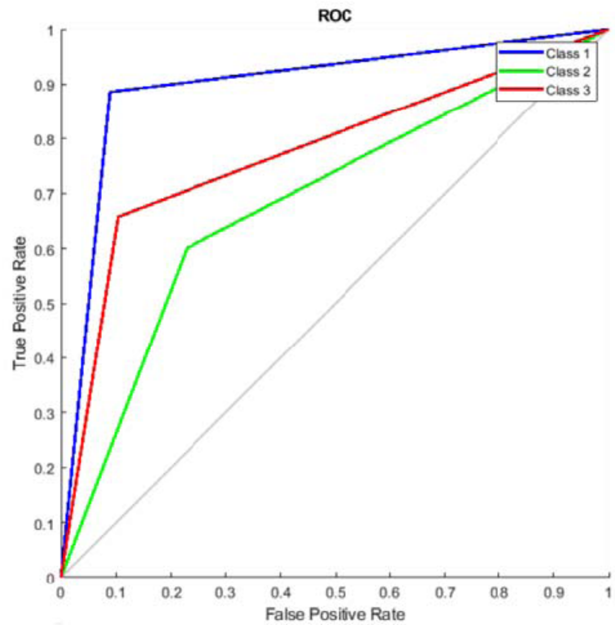

FIGURE 10. ROC for testing data flood sensor using ensemble (LSTM+RF).

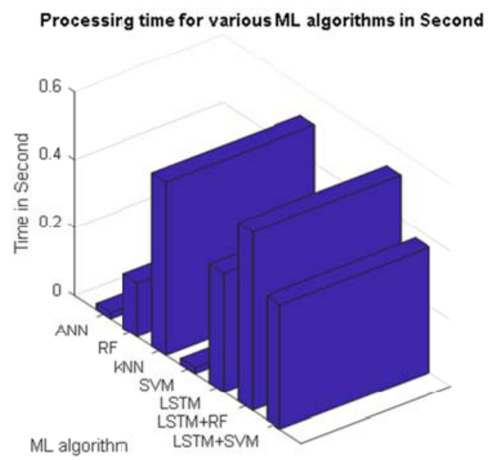

FIGURE 11. Processing time for individual and ensemble machine learning classifiers for testing.

network and random forest combined some kinds of soft nonlinear boundaries. In the evaluation performance testing sets, we applied 200 trees for our random forest model that were enough to receive smooth of separation. Eventually, the ensemble classifier generated an optimal and robust results and this was because individual classifiers suffer from overfitting because of dealing with the non-linearities in the data.

VII. CRITICAL DISCUSSION

In this paper, an efficient data science approach is developed and used to analyze 11 attributes related to flood sensor data from a total of 4214 records for detection of water level severity. The accuracy of ensemble LSTM+ RF classifier is 0.997 in training/validation phase, while an accuracy of 0.811 was shown in testing using unseen data. Figures 9 and 10 show the ROC of this classifier for training and testing, respectively. Furthermore, Figure 11 indicates that the ensemble classifiers have readable processing time in comparison to single classifiers during testing.

LSTM and random forest are two powerful classifiers for the analysis of data, which offer strong performance when compared to other models. These types of models applied the out-of-bag method based on the decision tree model rather than cross-validation to improve training and testing results.

In general, ensemble classifier preserves the appealing features of decision trees, such as dealing with irrelevant/redundant descriptors. In terms of the training procedure, this model was much faster compared to the ensemble classifier. A key reason that the ensemble classifier yielded higher accuracy is that it was able to generalize better by using combined evidence of its member classifiers. In spite of a number of samples in the data sets are mislabeled, the ensemble classifier can easily estimate the missing values, and work effectively with imbalanced data, which poses a challenge for other models, for instance, SVM.

As shown in Table 4, the ensemble classifiers generated high sensitivity and specificity for all the severity level classes while individual classifiers performed well using the sensitivity and specificity for the normal severity level class and failed to generate high performance to the other two classes (medium and dangerous level classes). This is due to the imbalanced representation of three classes in the dataset. However, the ensemble model performed comparatively better than individual models while simultaneously indicating superiority in the compromise between sensitivity and specificity for c2 and c3.

VIII. CONCLUSIONS AND FUTURE WORK

The collection of data through IoT platform and sensors mounted on the rivers can be used as inputs for the ML techniques to perform data science approaches for the detection of river flood severities. The proposed ensemble model in this research showed promising results for the detection of flood and can provide tools for warning for future flooding. Three flood data classes are considered in this research including normal, abnormal and dangerous water level classes. Performance evaluation metrics such as sensitivity and specificity and visualization techniques are used to evaluate the proposed ensemble machine learning approach. The results indicate that early warning of flood severity can be obtained using appropriate ensemble machine learning based data science techniques. Future work involves the use of particle swarm optimization and a genetic algorithm for optimization and selection of our machine learning approaches as well as the utilization of other deep learning algorithms for future regression of flood data.

REFERENCES

[1] Gartner. *Gartner Says 6.4 Billion Connected 'Things' Will Be in Use in 2017, Up 31 Percent From 2016*. Accessed: Feb. 27, 2020. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016,

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[3] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2923–2960, 4th Quart., 2018.

[4] J. Men, G. Xu, Z. Han, Z. Sun, X. Zhou, W. Lian, and X. Cheng, "Finding sands in the eyes: Vulnerabilities discovery in IoT with EUFuzzer on human machine interface," *IEEE Access*, vol. 7, pp. 103751–103759, 2019.

[5] A. Moraru, M. Pesko, M. Porcius, C. Fortuna, and D. Mladenic, "Using machine learning on sensor data," *J. Comput. Inf. Technol.*, vol. 18, no. 4, pp. 341–347, 2010.

[6] E. Borgia, "The Internet of Things vision: Key features, applications and open issues," *Comput. Commun.*, vol. 54, pp. 1–31, Dec. 2014.

[7] A. Y. Sun and B. R. Scanlon, "How can big data and machine learning benefit environment and water management: A survey of methods, applications, and future directions," *Environ. Res. Lett.*, vol. 14, no. 7, p. 28, 2019.

[8] U. K. Panchal, H. Ajmani, and S. Y. Sait, "Flooding level classification by gait analysis of smartphone sensor data," *IEEE Access*, vol. 7, pp. 181678–181687, 2019.

[9] G. Furquim, G. Filho, R. Jalali, G. Pessin, R. Pazzi, and J. Ueyama, "How to improve fault tolerance in disaster predictions: A case study about flash floods using IoT, ML and real data," *Sensors*, vol. 18, no. 3, p. 907, 2018.

[10] M. Bilal, A. Gani, M. Marjani, and N. Malik, "A study on detection and monitoring of water quality and flow," in *Proc. 12th Int. Conf. Math., Actuarial Sci., Comput. Sci. Statist. (MACS)*, Nov. 2018, pp. 1–6.

[11] K. Tuberson and K. Tremaine, "How the Internet of Things can prepare cities for natural disasters," Harvard Bussniss Review, 2017.

[12] I. Al Ridhawi, Y. Kotb, M. Aloqaily, Y. Jararweh, and T. Baker, "A profitable and energy-efficient cooperative fog solution for IoT services," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3578–3586, May 2020.

[13] S. Zafar, S. Jangsher, O. Bouachir, M. Aloqaily, and J. B. Othman, "QoS enhancement with deep learning-based interference prediction in mobile IoT," *Comput. Commun.*, vol. 148, pp. 86–97, Dec. 2019.

[14] F. Kreische and K. Zieman, "Foreword: The Internet of Things and its potential in international cooperation," Deutsche Gesellschaft für Internationale Zusammenarbeit, Bonn, Germany, Tech. Rep. 1, 2015.

[15] D. Dinculeană and X. Cheng, "Vulnerabilities and limitations of MQTT protocol used between IoT devices," *Appl. Sci.*, vol. 9, no. 5, p. 848, 2019.

[16] T. Khan, S. Ghosh, M. Iqbal, G. Ubakanma, and T. Dagiuklas, "RESCUE: A resilient cloud based IoT system for emergency and disaster recovery," in *Proc. IEEE 20th Int. Conf. High Perform. Comput. Commun., IEEE 16th Int. Conf. Smart City, IEEE 4th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Jun. 2018, pp. 1043–1047.

[17] N. Suri, Z. Zielinski, M. Tortonesi, C. Fuchs, M. Pradhan, K. Wrona, J. Furtak, D. B. Vasilache, M. Street, V. Pellegrini, G. Benincasa, A. Morelli, C. Stefanelli, E. Casini, and M. Dyk, "Exploiting smart city IoT for disaster recovery operations," in *Proc. IEEE 4th World Forum Internet Things (WF-IoT)*, Feb. 2018, pp. 458–463.

[18] Y. Kotb, I. Al Ridhawi, M. Aloqaily, T. Baker, Y. Jararweh, and H. Tawfik, "Cloud-based multi-agent cooperation for IoT devices using workflow-nets," *J. Grid Comput.*, vol. 17, no. 4, pp. 625–650, Dec. 2019.

[19] A. Sinha, P. Kumar, N. P. Rana, R. Islam, and Y. K. Dwivedi, "Impact of Internet of Things (IoT) in disaster management: A task-technology fit perspective," *Ann. Oper. Res.*, vol. 283, nos. 1–2, pp. 759–794, Dec. 2019.

[20] Z. Maamar, M. Sellami, N. Faci, E. Ugljanin, and Q. Z. Sheng, "Storytelling integration of the Internet of Things into business processes," in *Business Process Management Forum*. Cham, Switzerland: Springer, 2018, pp. 127–142.

[21] M. Al-Khafajiy, T. Baker, H. Al-Libawy, Z. Maamar, M. Aloqaily, and Y. Jararweh, "Improving fog computing performance via fog-2-fog collaboration," *Future Gener. Comput. Syst.*, vol. 100, pp. 266–280, Nov. 2019.

[22] S. Oueida, Y. Kotb, M. Aloqaily, Y. Jararweh, and T. Baker, "An edge computing based smart healthcare framework for resource management," *Sensors*, vol. 18, no. 12, p. 4307, 2018.

[23] D. B. Soubhagyalaxmi, L. S. Bhosle, T. Swetha, P. D. Veena, and B. V. Chetan, "Disaster management system using IoT," *Int. J. Res. Trends Innov.*, vol. 3, no. 6, pp. 148–154, 2018.

[24] R. Beltramo, P. Cantore, E. Vesce, S. Margarita, and P. De Bernardi, "The Internet of Things for natural risk management," in *Perspectives on Risk, Assessment and Management Paradigms*, A. Hessami, Ed. 2018, pp. 116–139.

[25] V. Lempitsky, M. Verhoek, J. A. Noble, and A. Blake, "Random forest classification for automatic delineation of myocardium in real-time 3D echocardiography," in *Proc. Int. Conf. Funct. Imag. Modeling Heart*. Berlin, Germany: Springer, 2009, pp. 447–456.

[26] L. Ohno-Machado, "Medical applications of artificial neural networks: Connectionist models of survival," Ph.D. dissertation, Stanford Univ., Stanford, CA, USA, 1996.

[27] M. Khalaf, A. J. Hussain, R. Keight, D. Al-Jumeily, R. Keenan, C. Chalmers, P. Fergus, W. Salih, D. H. Abd, and I. O. Idowu, "Recurrent neural network architectures for analysing biomedical data sets," in *Proc. 10th Int. Conf. Develop. eSyst. Eng. (DeSE)*, Jun. 2017, pp. 232–237.

[28] X. Chen, X. Zhu, and D. Zhang, "A discriminant bispectrum feature for surface electromyogram signal classification," *Med. Eng. Phys.*, vol. 32, no. 2, pp. 126–135, Mar. 2010.

[29] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[30] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, vol. 1, Aug. 1995, pp. 278–282.

[31] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[32] T. K. Ho, "A data complexity analysis of comparative advantages of decision forest constructors," *Pattern Anal. Appl.*, vol. 5, no. 2, pp. 102–112, Jun. 2002.

[33] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[34] V. Vapnik, *The Nature of Statistical Learning Theory*. Cham, Switzerland: Springer, 2013.

[35] P. R. Tools. (2013). *PRTools5 Introduction Version 5*. Accessed: Feb. 27, 2020. [Online]. Available: http://www.37steps.com/prtools/prtools5-intro/

[36] B. Fasel, "Robust face analysis using convolutional neural networks," in *Proc. Object Recognit. Supported User Interact. Service Robots*, 2002, pp. 40–43.

[37] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 1992, pp. 950–957.

[38] R. A. Jacobs, "Increased rates of convergence through learning rate adaptation," *Neural Netw.*, vol. 1, no. 4, pp. 295–307, Jan. 1988.

[39] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing* (NATO ASI Series), vol. 68, F. F. Soulié and J. Hérault, Eds. Berlin, Germany: Springer, 1990, pp. 227–236.

[40] K. Richmond, "Estimating articulatory parameters from the acoustic speech signal," Ph.D. dissertation, Univ. Edinburgh, Edinburgh, U.K., 2002.

[41] K. R. Kavitha, A. V. Ram, S. Anandu, S. Karthik, S. Kailas, and N. M. Arjun, "PCA-based gene selection for cancer classification," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res. (ICCIC)*, Madurai, India, Dec. 2018, pp. 1–4.

[42] S. Lawrence and C. L. Giles, "Overfitting and neural networks: Conjugate gradient and backpropagation," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 1, 2000, pp. 114–119.

[43] O. Fistikoglu and U. Okkan, "Statistical downscaling of monthly precipitation using NCEP/NCAR reanalysis data for tahtali river basin in turkey," *J. Hydrol. Eng.*, vol. 16, no. 2, pp. 157–164, Feb. 2011.

[44] Ö. Kişi, "Streamflow forecasting using different artificial neural network algorithms," *J. Hydrol. Eng.*, vol. 12, no. 5, pp. 532–539, Sep. 2007.

[45] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statist.*, vol. 46, no. 3, pp. 175–185, Aug. 1992.

[46] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[47] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proc. Int. Joint Conf. Neural Netw.*, 2000, pp. 189–194.

[48] J. OrShea, K. Crockett, W. Khan, P. Kindynis, A. Antoniades, and G. Boultadakis, "Intelligent deception detection through machine based interviewing," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[49] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994.

[50] W. Khan and K. Kuru, "An intelligent system for spoken term detection that uses belief combination," *IEEE Intell. Syst.*, vol. 32, no. 1, pp. 70–79, Jan. 2017.

[51] S. N. Alexandropoulos, S. B. Kotsiantis, and M. N. Vrahatis, "Data preprocessing in predictive data mining," *Knowl. Eng. Rev.*, vol. 34, p. e1, Jan. 2019.

[52] K. Kuru and W. Khan, "Novel hybrid object-based non-parametric clustering approach for grouping similar objects in specific visual domains," *Appl. Soft Comput.*, vol. 62, pp. 667–701, Jan. 2018, doi: 10.1016/j.asoc.2017.11.007.

[53] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[54] S. Jadhav, H. He, and K. Jenkins, "Information gain directed genetic algorithm wrapper feature selection for credit rating," *Appl. Soft Comput.*, vol. 69, pp. 541–553, Aug. 2018, doi: 10.1016/j.asoc.2018.04.033.

[55] R. A. Sheikh and A. Bhardwaj, "A review on integrating intrusion detection model using classifiers and chi-square feature selection," *Int. J. Advance Res., Ideas Innov. Technol.*, vol. 5, no. 5, pp. 230–233, 2019.

[56] I. S. Thaseen, C. A. Kumar, and A. Ahmad, "Integrated intrusion detection model using chi-square feature selection and ensemble of classifiers," *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 3357–3368, Apr. 2019, doi: 10.1007/s13369-018-3507-5.

[57] A. B. A. Christopher and S. A. A. Balamurugan, "Prediction of warning level in aircraft accidents using data mining techniques," *Aeronaut. J.*, vol. 118, no. 1206, pp. 935–952, Aug. 2014.