



FLOOD PREDICTION AND ASSESSMENT PLATFORM A MULTI-MODEL APPROACH

Dr. Elizabeth Isaac
Assistant Professor
Dept of Computer Science
and Engineering
Mar Athanasius College Of
Engineering.
Kothamangalam, India

Aravind Balakrishnan
Dept of Computer Science
and Engineering.
Mar Athanasius College Of
Engineering.
Kothamangalam, India

Jiju S Jacob
Dept of Computer Science
and Engineering.
Mar Athanasius College Of
Engineering.
Kothamangalam, India

Nandu Viswanathan
Dept of Computer Science
and Engineering
Mar Athanasius College Of
Engineering.
Kothamangalam, India

Abstract— Floods are the most destructive natural disasters, causing huge damage to life and socioeconomic system. Early flood warning systems are good countermeasures against flood hazards. So here we propose a flood prediction system that uses machine learning models to deliver chances of flood based on location and time queries. The system can be integrated with multi-model deep learning methods such as Support Vector machines, Neural Networks and Extreme Learning Model to predict the chances of flood based on various environmental, climatic factors and geographical conditions. Based on the results, actions that need to be taken can also be predicted. The model is trained on a comprehensive data set expanding over years. The system has the ability to compare the results of various sub models such that the overall accuracy can be modified. It is helpful for the people living in the flood affected areas as well as to the ones who are related to the organizations in this field.

Keywords—flood, multi-model, neural network, support vector machine, Extreme learning machines, Decision Tree Classifier

I. INTRODUCTION

Floods are a major concern in India due to the variety of climate and landscapes across India, it is a tremendous task to prevent and predict floods. As the cause of floods change according to terrain, precipitation, climate and other numerous attributes an effective system that can predict and assess the impact of flood is still a far fetched dream. Though various approaches has been used to measure one of the most deadliest natural disaster a near to accurate model has yet to be announced. Floods as it causes damages to life, infrastructure, and agriculture an accurate methodology to prevent flood and take necessary actions to reduce the loss are

high. This paper utilizes machine learning methods such as Support vector Machines(SVM), Neural networks to predict the occurrence as well as the severity it causes by analyzing various attributes including terrain, quarter of year, precipitation and climate. As SVM have been proven to yield accurate results over a small dataset, it is quite powerful when encountering a complex problem such as flood prediction as the data availability is a major concern in the field. SVM has been used in other fields including statistics, management and economics and have been proven to be effective. While neural networks provide with better results over a wide span of data. The multi-model approach compares the result of the models thus providing a better analytical view.

II. RELATED WORKS

A. Support Vector Machine

SVM is a supervised machine learning algorithm which can be used for classification or regression problems. It uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs. SVM [1] was introduced as a class of support vector, used to minimize over-fitting and reduce the expected error of learning machines. SVM is greatly popular in flood modelling; it is a supervised learning machine which works based on the statistical learning theory and the structural risk minimization rule. SVMs are suitable for both linear and non-linear classification problems. SVM is used to predict a quantity forward in time based on training from past data. Over the past two decades, the SVM was also extended as a regression tool, known as support vector regression (SVR).

SVM algorithm is based on the concept of decision planes. It finds lines or boundaries that correctly classify the training data set. Then, from those lines or boundaries, it picks the one

that has the maximum distance from the closest data points. When straight lines can't be used to classify, we use kernel [12] SVM which projects non-linear separable data sets of lower dimension to linearly separable data sets of higher dimension.

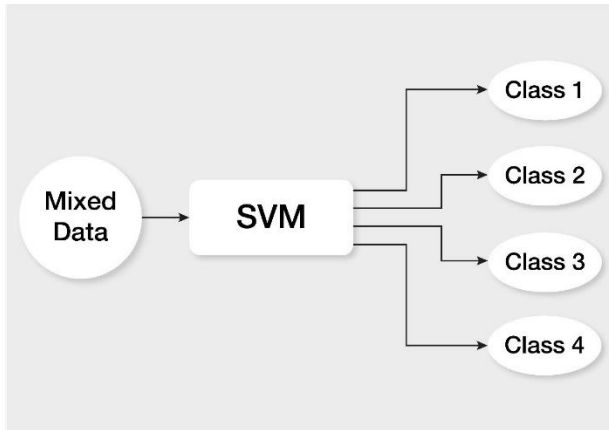


Fig 2.1 SVM Classifier

We consider data points of the form $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_n, y_n)$. where $y_n = 1/-1$, a constant denoting the class to which point x_n belongs, $n =$ number of sample, each x_n is a p -dimensional

real vector. We take $w \cdot x + b = 0$ —(1)

where b is scalar and w is a p -dimensional vector. The parallel hyperplanes can be described by the equation, $w \cdot x + b = 1$
 $w \cdot x + b = -1$

By geometry, the distance between the hyperplanes is $2/w$. So we want to minimize w . Therefore,

$$y_i (w \cdot x_i - b) \geq 1; 1 \leq i \leq n$$

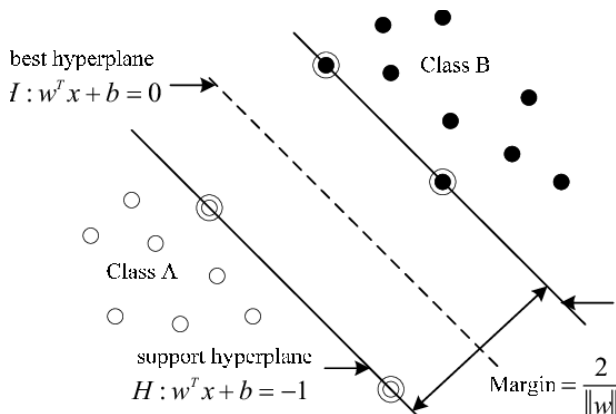


Fig 2.2 Maximum margin hyperplanes for a SVM trained with samples from two classes

SVMs fall into the intersection of two research areas [4]: kernel methods, and large margin classifiers. SVM has been applied to feature selection, time series analysis, reconstruction of a chaotic system, and non-linear principal components. We can classify linearly separable and non-linear separable data using Support Vector Machine.

1) *Linear Classification*: The SVM analysis attempts to find a 1-dimensional hyperplane (i.e. a line) that separates the cases based on their target categories. There are an infinite number of possible lines. An SVM analysis finds the line(or, in general, hyperplane) that is oriented so that the margin between the support vectors is maximized.

2) *Non Linear Classification*: If the points are separated by a non-linear region, SVM handles this by using a kernel function to map the data into a different space where a hyper plane can be used to do the separation.

The kernel function may transform the data into a higher dimensional space to make it possible to perform the separation. Kernel functions are a class of algorithms for pattern analysis or recognition, whose best known element is the support vector machine (SVM).

The Linear kernel is the simplest kernel function. It is given by the inner product plus an optional constant c . The default and recommended kernel function is the Radial Basis Function (RBF). The RBF kernel nonlinearly maps samples into a higher dimensional space unlike to linear kernel. The RBF kernel has less hyper parameters than the polynomial kernel.

B. Neural Networks

An Artificial Neuron [17] is basically an engineering approach of biological neuron. It has device with many inputs and one output. ANN [3] is consisting of large number of simple processing elements that are interconnected with each other and layered also. A neural network is used for various objectives such as clustering, functional approximation, prediction, optimization etc. The paper discusses about two types of neural networks, Extreme Learning Machine(ELM) [2] [14] and back propagation neural network [18]. The key difference between the two models is that in ELM is a feed forward neural networks for classification, regression, clustering, sparse approximation, compression and feature learning with a single layer or multiple layers of hidden nodes, where the parameters of hidden nodes (not just the weights connecting inputs to hidden nodes) need not be tuned. These hidden nodes can be randomly assigned and never updated. In case of a back propagated neural network the weights are updated according to the error correction function.

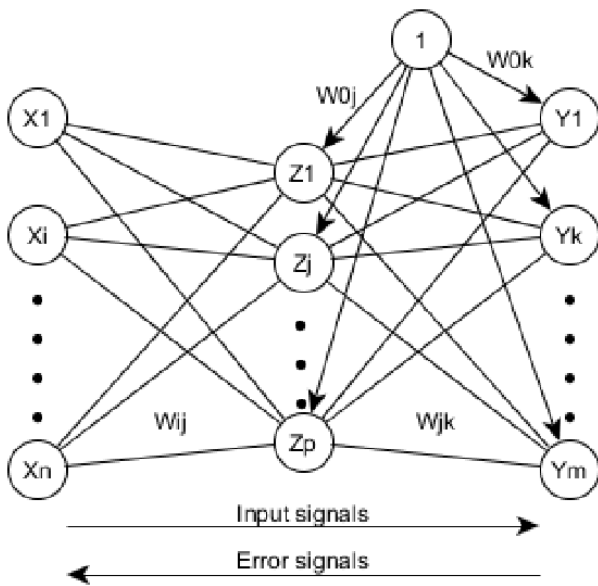


Fig 2.3 Back propagation structural diagram

Extreme learning machine (ELM) is a new learning algorithm for the single hidden layer feed forward neural networks. Compared with the conventional neural network learning algorithm it overcomes the slow training speed and over-fitting problems. ELM is based on empirical risk minimization theory and its learning process needs only a single iteration. The algorithm avoids multiple iterations and local minimization. It has been used in various fields and applications because of better generalization ability, robustness and fast learning rate.

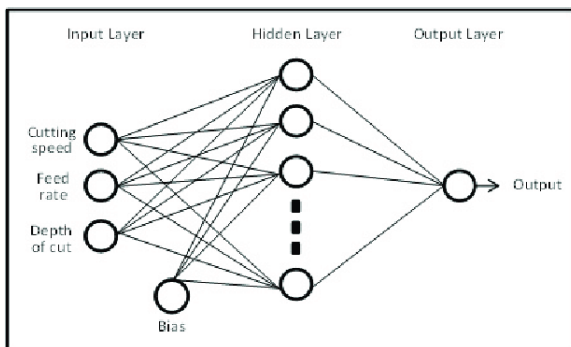


Fig 2.4 Extreme Learning Machine structure

The Back propagation algorithm [18] looks for the minimum value of the error function in weight space using a Technique called the delta rule or gradient descent. The weights that minimize the error function is then considered to be a solution to the learning problem. The application of the generalized delta rule involves two phases. At the first phase, to compute the output values for each output unit, the input is presented and propagated forward through the network. Desired output value do is compared with this output, for each output unit resulting in an error signal. To calculate the appropriate weight changes, a backward pass is involved by

the second phase through the network during which the error signal is passed to each unit in the network. Fig 5 shows the flowchart of the training of the neural network with Backpropagation algorithm and Fig 4 shows the structure of the ANN with respect to back propagation.

C. Decision Tree

It is flow-chart like tree structure, where each internal node denotes a test on an attribute, each branch denotes an outcome of test, and each leaf node holds a class label. The topmost node in a tree is the root node [5]. Given a tuple, the attribute values of the tuple are tested against decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that tuple. Decision tree is useful because construction of decision tree classifiers does not require any domain knowledge. It can handle high-dimensional data. The learning and classification steps of decision tree induction are simple and fast. Their representation of acquired knowledge in tree form is easy to assimilate by users. Decision tree classifiers have good accuracy [6]. The major decision tree advantages are transparency, specificity, resilience, validation.

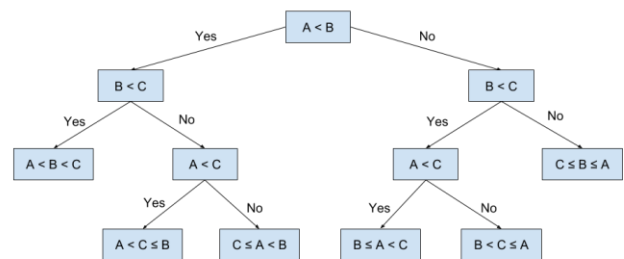


Fig 2.5 Decision Tree

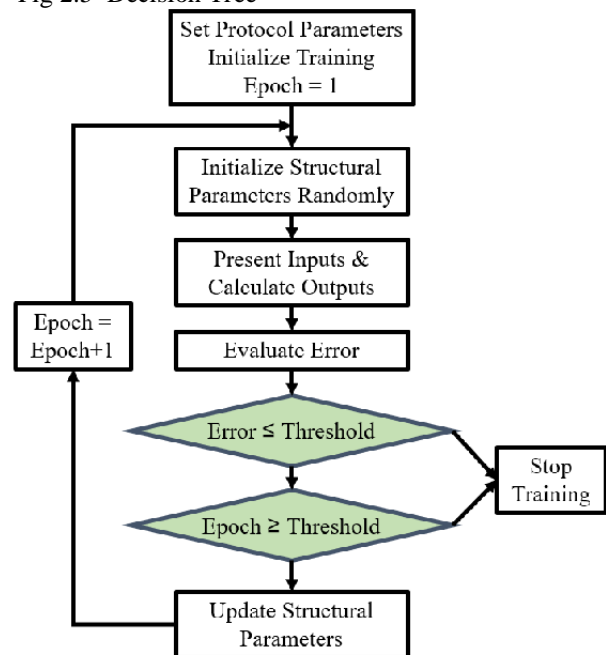


Fig 2.6 Back propagation training algorithm



Decision Tree [16] algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data). In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

III. SYSTEM DESIGN

A. Components

- A python flask based web interface to provide communication between the trained machine learning model and the user to input custom data for prediction
- Support Vector Machine model
- Artificial Neural Network
- Decision Tree Classifier

B. Interface

A python flask [19] based web interface to input location and time to predict the flood level. Flask is a python based lightweight web framework which can easily call trained machine learning models via request API calls. The platform uses virtual earth API to get the longitude and latitude of the inputted location and *openweathersmap* API is used to get the approximate precipitation data of the location at the inputted time. The server routes the data to the ML models and returns the result of each model in a six-max scale where 0 being the weakest flood and 6 being the extremely severe flood. The inputted region is first classified using country code such as "IN". The location is further pin-pointed into state-wise. According to the later classification the region is classified into Island, Coastal, Hilly etc. The date is chosen using the *JS date-picker* API. Further, the date rounded into seasons and the end result is passed into the model with the highest accuracy in this case the ANN.

C. Information

A	B	C	D	E	F	G	H	I	J
SUBDIVISION	YEAR	ANNUAL PRECIPITATION	QUARTER PRECIPITATION	SEVERITY	TERRAIN	CLIMATE	SEVERITY		
ANDAMAN & NICOBAR ISLANDS	1980	3625.3	Jan-Feb	6.3	0 Island	Tropical	0		
ANDAMAN & NICOBAR ISLANDS	1990	3625.3	Mar-May	776.5	0 Island	Tropical	0		
ANDAMAN & NICOBAR ISLANDS	1980	3625.3	Jun-Sep	1946.3	4 Island	Tropical	4		
ANDAMAN & NICOBAR ISLANDS	1980	3625.3	Oct-Dec	896.3	0 Island	Tropical	0		
ANDAMAN & NICOBAR ISLANDS	1981	3074.8	Jan-Feb	38.9	0 Island	Tropical	0		
ANDAMAN & NICOBAR ISLANDS	1981	3074.8	Mar-May	366.7	0 Island	Tropical	0		
ANDAMAN & NICOBAR ISLANDS	1981	3074.8	Jun-Sep	1908	0 Island	Tropical	4		
ANDAMAN & NICOBAR ISLANDS	1981	3074.8	Oct-Dec	761.2	0 Island	Tropical	0		

Fig 3.1 Data set structure

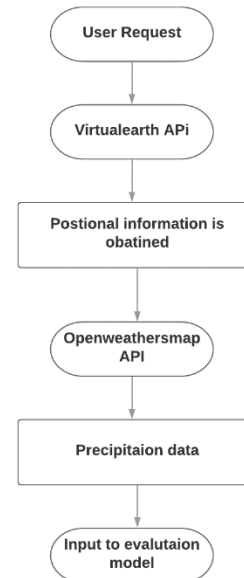


Fig 3.2 Information flow

The data set in the study which is depicted in Fig 3.1 contains subdivision of the Indian continent, annual and quarter precipitation data, year, terrain and the corresponding severity of flood caused by these parameters. Information is stored in the form of tables and the live information regarding the precipitation is collected from the *openweathersmap* API. For the API to work it needs information regarding the positional attributes such as longitude, latitude it is obtained from *virtualearth* API which is a part of *BingMaps* [20] [21] *Representational State Transfer*(REST) [22] API service. The *virtualearthapi* passes information in the format longitude, latitude, city and state. The platform acquires location and date from the user and passes it to *BingMaps* API and details such as city, latitude, longitude are obtained it is further passed to *openweathersmap* API and precipitation data about the location is obtained ultimately these data are passed to model for evaluation. The flow of information is represented in Fig 3.2.

D. Design flow

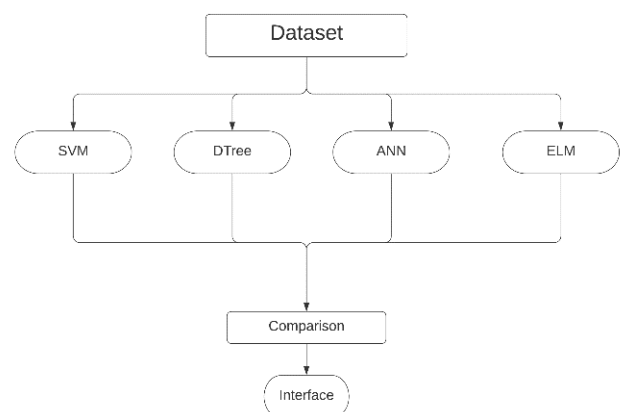


Fig 3.3 Platform flow



The various models are trained in the dataset provided and the accuracy is compared and the model with the highest accuracy in this case the ANN is linked to the interface as an API call . The dataset is normalized for each model as in the case of SVC the dataset containing categorical data is mapped to numerical data using Label encoders which is a part of *scikit* python library and to avoid the relation between these newly encoded categorical data such as the regions *onehotencoders* [9] are used to further encode the data set such that the model can handle categorical data. As normally machine learning models can't handle categorical data. This normalization is applied for every models in study. Fig 3.3 shows the overall working of the platform with where the data set is provided to each model and the result is compared and the best model is linked with the interface.

IV. MODEL EVALUATION

The model in study includes Support Vector Machine (SVM), Decision Tree, Artificial Neural Network, Extreme Learning Machines. All these machine learning models were trained on the same data set with the same type of normalization applied. The Neural network proved to be the most accurate in for the given data set considering Extreme Learning Machines have the same configuration as the Neural network with three layers including two hidden layers with optimizer as "adam" [7]. The batch size was 10 and the epoch is 500. The common configuration is depicted in Fig. 4.1 with three dense layers including the two hidden and the input layer. The models were rated in the range of 96 to 94 percent in terms of accuracy.

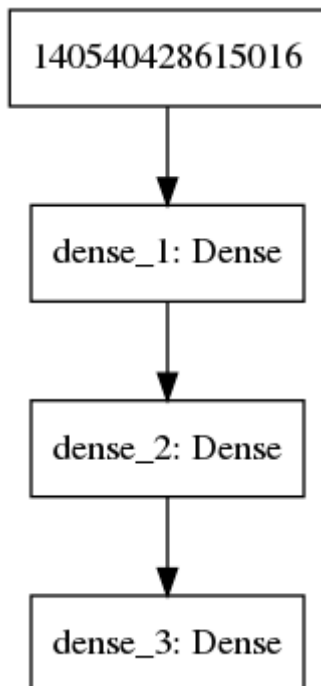


Fig 4.1 Network configuration

V. FUTURE WORK

The dataset used in this multi-model approach considers mainly terrain category, quarter, precipitation, climate. The prediction of criteria can be extended to other attributes such as soil data, attributes of water bodies, wave data etc. The integration of all these data and retraining these model in dynamic DAG framework such *pytorch* [9] will yield much accurate results.

All the machine learning models specified in the paper are independent, if modular approach is applied to these models by utilizing an over looker the accuracy can be improved as the workload is distributed with among these models and the comparison can be implemented in prediction rather than in the training phase. But the scope of the mentioned is beyond this paper.

The dataset in study contains only text data and the accommodation of other types of data such as satellite images etc can increase the parameters involving the prediction of floods.

VI. CONCLUSION

The prediction and assessment of floods especially in a geographic scene such as India is a tremendous task and requires high amount of data and processing. The acquiring of data is trivial in case flood prediction and the platform accuracy is improved when the multiple models are incorporated instead of relying on a single machine learning model.

VII. REFERENCES

- [1] Yujun Yang, Jianping Li and Yimei Yang, "The research of the fast SVM classifier method," 2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing (IC-CWAMTIP), Chengdu, 2015, pp. 121-124.
- [2] Guang-Bin Huang, Qin-Yu Zhu and Chee-Kheong Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541), Budapest, 2004, pp. 985-990 vol.2.
- [3] H. Xia and Q. Rao, "Flood Level Prediction on the Basis of the Artificial Neural Network," 2009 First International Conference on Information Science and Engineering, Nanjing, 2009, pp. 4887-4890.
- [4] Himani Bhavsar, Mahesh H. Panchal "A Review on Support Vector Machine for Data Classification," International Journal of Advanced Research in Computer Engineering Technology (IJARCET) Volume 1, Issue 10, December 2012.
- [5] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu and Philip S. Yu, et al., "Top 10 algorithms in data mining", Knowledge and



- Information Systems, Vol. 14, No. 1, 1-37, DOI: 10.1007/s10115-007-0114-2.
- [6] Hang Yang, Fong, S, "Optimized very fast decision tree with balanced classification accuracy and compact tree size," Data Mining and Intelligent Information Technology Applications (ICMiA), 2011 3rd International Conference on, Pp.57-64, 24-26 Oct. 2011.
- [7] Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks," 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 2018, pp. 1-2.
- [8] C. Heghedus, A. Chakravorty and C. Rong, "Neural Network Frameworks. Comparison on Public Transportation Prediction," 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Rio de Janeiro, Brazil, 2019, pp. 842-849.
- [9] V. Patel K.S. and K. S. Gurumurthy, "Design of High Performance Quaternary Adders," 2011 41st IEEE International Symposium on Multiple-Valued Logic, Tuusula, 2011, pp. 22-26.
- [10] A. Shrestha and A. Mahmood, "Review of Deep Learning Algorithms and Architectures," in IEEE Access, vol. 7, pp. 53040-53065, 2019.
- [11] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, July-Aug. 1998.
- [12] A. Patle and D. S. Chouhan, "SVM kernel functions for classification," 2013 International Conference on Advances in Technology and Engineering (ICATE), Mumbai, 2013, pp. 1-9.
- [13] T. Baltrušaitis, C. Ahuja and L. Morency, "Multimodal Machine Learning: A Survey and Taxonomy," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 2, pp. 423-443, 1 Feb. 2019.
- [14] H. Ke and X. Huang, "An improved Extreme Learning Machine," Proceedings of the 32nd Chinese Control Conference, Xi'an, 2013, pp. 3232-3237.
- [15] A. Navada, A. N. Ansari, S. Patil and B. A. Sonkamble, "Overview of use of decision tree algorithms in machine learning," 2011 IEEE Control and System Graduate Research Colloquium, Shah Alam, 2011, pp. 37-42.
- [16] F. Yuan, F. Lian, X. Xu and Z. Ji, "Decision tree algorithm optimization research based on MapReduce," 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, 2015, pp. 1010-1013.
- [17] M. Mishra and M. Srivastava, "A view of Artificial Neural Network," 2014 International Conference on Advances in Engineering Technology Research (ICAETR - 2014), Unnao, 2014, pp. 1-3.
- [18] Hecht-Nielsen, "Theory of the backpropagation neural network," International 1989 Joint Conference on Neural Networks, Washington, DC, USA, 1989, pp. 593-605 vol.1.
- [19] M. R. Mufid, A. Basofi, M. U. H. Al Rasyid, I. F. Rochimansyah and A. rokhim, "Design an MVC Model using Python for Flask Framework Development," 2019 International Electronics Symposium (IES), Surabaya, Indonesia, 2019, pp. 214-219.
- [20] X. Qu et al., "A spatial web service client based on Microsoft Bing Maps," 2011 19th International Conference on Geoinformatics, Shanghai, 2011, pp. 1-5.
- [21] L. Bećirspahić and A. Karabegović, "Web portals for visualizing and searching spatial data," 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, 2015, pp. 305-311.
- [22] S. M. Sohan, F. Maurer, C. Anslow and M. P. Robillard, "A study of the effectiveness of usage examples in REST API documentation," 2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Raleigh, NC, 2017, pp. 53-61.