An Evolutionary MultiLayer Perceptron Algorithm for Real Time River Flood Prediction

Geerish Suddul[1], Kumar Dookhitram[1], Girish Bekaroo[2], Nikhilesh Shankhur[1]
[1]University of Technology, Mauritius (UTM)
[1]Pointe-aux-Sable, Mauritius
[2]Middlesex University Mauritius
[2]Flic en Flac, Mauritius
g.suddul@umail.utm.ac.mu

***Abstract*— Severe flash flood events give very little opportunity for issuing warnings. In this paper, we approach the automated and real time prediction of river flooding by proposing and evaluating different variations of the conventional Multilayer Perceptron (MLP) machine learning algorithm. Our first approach follows a trial and error attempt to optimize the MLP architecture. The second and third approaches are based on the application of nature inspired evolutionary techniques, namely the Genetic Algorithm (MLP-GA) and the Bat Algorithm (MLP-BA) respectively. The MLP-GA generates an improved MLP configuration and MLP-BA enhances the training method. Our fourth, novel approach (MLP-BA-GA) is based on the application of GA to further optimize both the BA and MLP architecture. When compared with previous work, experiments show improvement in the accuracy of river flood prediction, with significant results for the MLP-BA-GA.**

Keywords— Artificial Neural Network; Bat Algorithm; Flood Prediction; Genetic Algorithm; Machine Learning Optimization; Metaheuristic Model, Evolutionary Computing.

I. INTRODUCTION

Global warming is responsible for changes in precipitation patterns causing severe weather conditions such as torrential rainfall and flood. Several regions around the globe deal with the sudden rise of water level causing damage to infrastructure and the loss of human life. This situation, known as flash flood, is even more important for less developed and developing countries, causing drastic impacts on their economy. On a worldwide scale from 1980 to 2009, it is estimated that flood was responsible for more than 500,000 loss of life and affected more than 2.8 billion people [1].

Among different approaches used to monitor floods, we focus on the use of Wireless Sensor Networks (WSN). Different parameters, such as water level, temperature, radiation can be monitored in near real time. This large set of collected data is analyzed to forecast the possible occurrence of a flood and such predictions can be lifesaving. It is therefore important to automate this process in an attempt to minimize human intervention and avoid unnecessary delays. Several Machine Learning (ML) algorithms have been used towards this end [2, 3], as they can analyze the collected data, which have been pre-processed, and generate learning models to make forecasts autonomously. It has been demonstrated by Furquim et al. [2] that traditional Machine Learning techniques, such as Bays Net, MLP, J48, Random Forest, Random Tree and BF Tree provide different level of prediction accuracy. Some of which are quite appropriate for flood prediction, but more recent studies, in different other domains, using hybrid techniques provide even better results [7, 8, 9] than standalone approaches.

We believe that further optimization of Machine Learning techniques, especially Artificial Neural Networks (ANNs) is possible for better prediction of flash flood nowcasting with the combination of Genetic Algorithm (GA) and nature inspired techniques. An ANN emulates the human brain consisting of different layers such as the input layer, processing (hidden) layer(s) and the output layer. The layers are made up of nodes (neurons) connected by weighted edges (synapses). Feed-forward propagation is the process by which data are fed in from input nodes to the output nodes. Back propagation is the learning process by which the output error of the network is propagated backwards through the network and weights are adjusted. GAs are typically used to optimize ANNs and are inspired by the natural selection process, where the fittest individuals are selected for producing offspring for the next generation. The network parameters for the offspring are inherited from the fittest parent network in the previous generation. Swarm based nature inspired algorithms such as the Bat Algorithm (BA) is an emerging optimization approach which can further enhance the prediction accuracy of Machine Learning techniques. BA is based on the echolocation of micro-bats which allows them to collectively recognize tiny preys among various existing living and non-living objects, in night darkness [10]. It therefore combines capabilities of both population-based and local search techniques. With its strong local search ability, the algorithm can converge to a global optimum, enhancing machine training aspects like feature selection, hidden layers and number of nodes per layer [11, 12].

In this research work, we propose to analyze and evaluate the following Multi-Layer Perceptron (MLP) metaheuristics hybrid models: Optimized MLP, MLP-GA, MLP-BA and MLP-BA-GA.

II. LITETRATURE REVIEW

C.L. Wu and K.W. Chau [3] present the comparison of an Artificial Neural Network with Genetic Algorithm (ANN-GA) with Adaptive-Network-Based Fuzzy Inference System (ANFIS) against Linear Regression (LR) in flood forecasting. The ANFIS model performs well in different environments and also accomplishes faster convergence. As for the ANN-GA, it is configured with a three-layer network, three input nodes and one output node. The data has been normalized between 0 and 1. The three nodes in the hidden layer is defined as optimal through trial and error. It has been demonstrated that the ANFIS model yields higher accuracy but at higher computational cost. Comparison of the three models over a 24-hour lead time shows that the absolute error is more considerable for Linear Regression model and much smaller with the ANFIS model, which also requires less training time than ANN-GA model. However, ANFIS model requires more parameters than the other models.

A modified Bat Algorithm to optimize the weights and structure of ANNs is proposed by Jaddi et al [4]. Along with a standard Bat Algorithm, two modified versions (MBatDNN and MeanBatDNN) are proposed as an attempt to improve the search strategy of the algorithm in the population. Following their application to both classification and time series problems (rainfall data), the LogisticBatDNN has proved to be more efficient. Further experiments using Taguchi method to tune the parameters of LogisticBatDNN showed even better performance. It is concluded that with regards to classifications error, T-LogisticBatDNN algorithm is a better approach compared to the two algorithms.

Kora & Kalva [5] focus on the application of Bat Algorithm in the detection of myocardial infarction. BA and IBA (Improved Bat Algorithm) have been used mainly for feature selection from ECG data while ignoring noise and redundancy. The proposed IBA is less prone to blocking at a local minimum and thus converges to a lower global minimum. The processed inputs have been applied to the following classification algorithms: SVM, KNN, LM NN, SCG NN. LM NN with IBA demonstrating highest accuracy of 98.9% compared to the standard BA with only 58.7%.

Several meta heuristic algorithms and clustering methods have been evaluated for the diagnosis of heart disease by Roostaee & Ghaffary [6]. Binary Cuckoo Optimization Algorithm (BCOA) is proposed for feature selection and Support Vector Machine (SVM) is used for constructing the model. Experiments demonstrate that their approach are optimal, as compared to other hybrid algorithms. The model achieved an accuracy of around 84.44%, a sensitivity near 86.49%, and a specificity of about 81.49%.

III. APPROACH

A. Machine Learning Approach

Our approach consists of an optimized MLP algorithm and three MLP hybrid techniques as described below:

Optimized MLP is configured with 2 layers each with 32 neurons. Backpropagation is used to train the model and early stopping ensures that the training is optimal by avoiding overfitting. To further prevent overfitting, dropout of 20% at first layer is configured. Dropout refers to ignoring a percentage of neurons which hinders specific neurons from proper adaption and thus, allowing all neurons to train equally.

MLP-GA is configured to build 15 networks with different parameters for each generation. After back propagation, only the five best networks are chosen. The next generation then inherits the configurations from these five best networks and new random parameters are added to the new generation networks. At completion of all generations, the most appropriate configuration is derived and applied to the prediction model in order to yield the best accuracy.

MLP-BA has a configuration of 3 layers each with 20 neurons for MLP and BA with 100 generations (iterations) each consisting of 100 bat population. For each generation 100 different combination of networks weights and biases are generated. The cost of each 100 networks are computed. The bat with the lowest cost is selected for weights inheritance of the next generation. The sequence is then repeated for the 100 generation to find the network global minimum cost.

MLP-BA-GA. GA is used to train both the network architecture and BA configurations. BA keeps generating the network weights for a number of populations and iterations until the lowest cost is found by the MLP. Five network architecture configurations and BA configurations with the lowest cost are selected. The next GA generation inherits the configurations of the previous best and the iteration starts again until all GA generations completes. After the network is trained, the testing set is used for prediction. The steps are depicted in Figure 1.

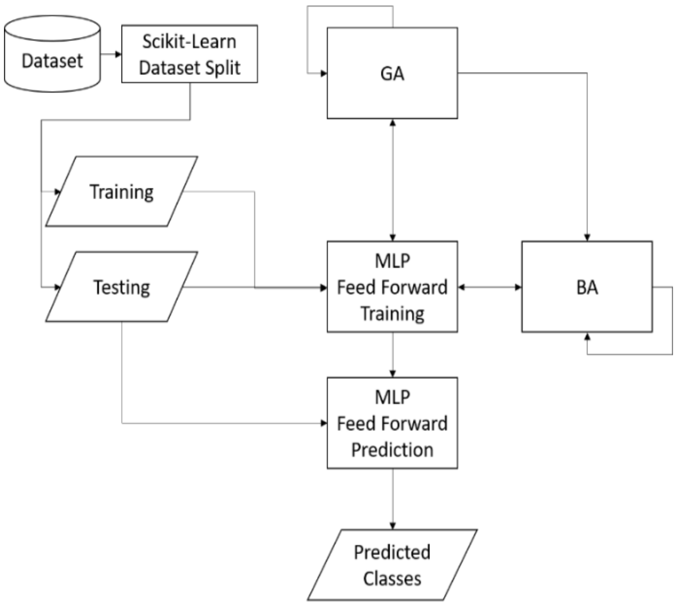

Fig. 1. MLP-BAT-GA Process Flow

B. Data Set and Data Preparation

The data set corresponds to predefined categories of water level collected by a WSN for a river in Brazil available from Furquim et al. [2]. All attributes are scaled in real numbers between 0 and 1 and consists of two main input types, namely raw previous data and specific attributes of the raw data in window sizes of 5 and 10 previous readings. The following vectorized attributes of the raw data have been considered, namely average, median, standard deviation, the sum, and the difference between the last and the first values.

We ensure that the dataset is shuffled and split into 75% training set and 25% test set for validation. A random state of 66 is used to ensure that each experiment splits the dataset with same record in every set. Therefore, ensuring an appropriate calculation of prediction accuracy for each model.

IV. EVALUATION

A. Experimental Setup

We develop our models using Python packages (Scikit-Learn, Tensor Flow & Keras). Scikit-Learn is used to read and split the dataset for training, testing, scale values and design and run the ML algorithms. TensorFlow is used to design, train and execute the Artificial Intelligence models, as well as to calculate accuracy. Keras provides the possibility to rapidly build prototypes and runs on CPU and GPU. The hardware configuration for all our experiments is as follows: CPU: Intel Core i7-7700HQ @ 3.80 GHz with 4 cores and 4 threads, GPU/MEM: NVIDIA GTX 1050, 768 CUDA cores @ 1645 MHZ/4 GB GDDR5, RAM: 8 GB DDR4 and Hard Disk: SSD.

B. Results & Discussion

Following an evaluation of the models, the results are presented in Figure 2 with 5 previous readings (window size 5) and Figure 3 (window size 10). The accuracy represents the height of water level and takes the form of an average of three-time step prediction for both raw data and attributes of the data, and is calculated as per the formula below.

$$Prediction\ Accuarcy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ Predictions\ Made}$$

For comparison purposes, we have included the best results obtained from the MLP algorithm used by Furquim et al [2] denoted as MLP (Furquim). According to Figure 2 the MLP-BA-GA scored the highest accuracy of around 80.47% and 78.86% for raw data and attributes of the data, respectively. It depicts an increase of 7.87% and 4.53% when compared with traditional approaches such as results of MLP (Furquim) and 6.08% and 3.65% when compared with our optimized MLP. For window size of 10 seconds, according to Figure 3 the MLP-BA-GA provides an increase in accuracy level as compared to all the different models.

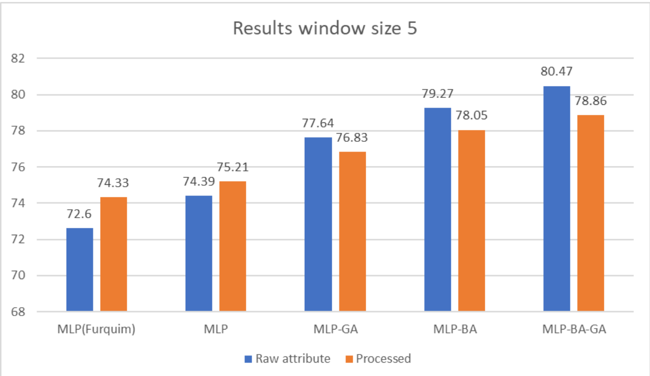

Fig. 2. Prediction Accuracy (w=5)

The proposed MLP-BA-GA accuracy obtained for both window size 5 and 10 is higher compared to other techniques. When compared to MLP-BA it can be noticed that the GA accounts for an increase in accuracy. It's mainly responsible

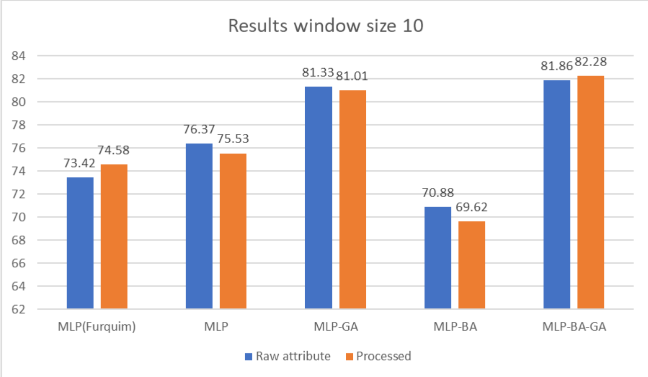

Fig. 3. Prediction Accuracy (w=10)

for creating and optimizing the network architecture (e.g. layers, neurons) and selecting the best BA configuration altogether to obtain the lowest cost and highest accuracy by the network. The highest increase has been registered when compared with MLP-BA based on 10 previous records (w=10). This is a particular case as there is a significant drop in accuracy between w=5 and w=10, as can be depicted by comparing MLP-BA results across Figure 2 and Figure 3. We have noted that the current MLP configuration with BA is overfitting the dataset with 85.65% in training accuracy and 63.29% in test accuracy. The Bat Algorithm is also missing the global minimum for the test data and converging more towards a low loss for the train data. As a result, implementing the Genetic Algorithm for MLP-BA, helps to generate a more appropriate configuration and bring considerable improvement in accuracy.

In an attempt to further evaluate the application of our approach, we measured the average computational time for each experiment. Both the training time and prediction time are presented in Table 1 below.

Table 1: Computational Time (Seconds)

Hardware	Optimized MLP	MLP-GA	MLP-BA	MLP-BA-GA
Training	10.91	175.94	58.54	13423.51
Prediction	0.03	0.25	0.02	0.35

Measurements regarding the average computational time for each experiment demonstrates that the training time for the MLP-BA-GA is quite significant, as it requires a large amount of computer resources. Once the training is complete though, the prediction time is almost comparable to all the other algorithms, as the same MLP process is applied. Therefore, our deployment model follows an offline training approach, which implies that training sessions will not hinder on the operation of a live running model. Following every offline training session with new data, the updated generated model can then be deployed. Nevertheless, the frequency of training will greatly depend on the occurrences of new flash flood events, which is region specific. As such, in cases where frequent and online training is required, MLP-BA with window size of 5 and MLP-GA with window size of 10 can be considered respectively. They have also demonstrated significant degree of improvement although less that MLP-BA-GA.

V. Conclusion & Future Works

The results demonstrate that our hybrid Metaheuristic Machine Learning algorithm (MLP-BA-GA) increases the accuracy of prediction results. It is of main importance for time critical tasks, especially when human lives are at stake due to natural calamities like river floods. The MLP-BA-GA is recommended in cases where training can be performed offline, as this process imposes an intensive use of computer resources and is time consuming. Further improvement can be achieved by implementing a feature selection algorithm to filter out unnecessary features. It has the potential to further improve accuracy as well as reduce the computation complexity which in turns lowers the computation time. Also, an investigation of different set of window sizes and data attributes can also contribute to improve the nowcasting accuracy. From a deployment perspective, the proposed approach can take the form of a web service with a REST interface. As such, mobile and web applications can be developed and receive updates with regards to possibilities of river overflowing. It can benefit communities which lives close to rivers as well as the necessary authorities responsible for dealing with natural calamities.

References

[1] S. Doocy, A. Daniels, C. Packer, A. Dick and T. Kirsch, "The human impact of earthquakes: a historical review of events 1980-2009 and systematic literature review," PLoS currents, 2013.

[2] G. Furquim, F. Neto, G. Pessin, J. Ueyama, P. Joao, M. Clara, E. Mendiondo, V. De Souza, P. De Souza, D. Dimitrova and T. Braun, "Combining wireless sensor networks and machine learning for flash flood nowcasting," Advanced Information Networking and Applications Workshops (WAINA) IEEE, pp. 67-72, 2014.

[3] C. Wu and K. Chau, "Evaluation of several algorithms in forecasting flood," Berlin, Heidelberg, 2006.

[4] N. Jaddi, S. Abdullah and A. Hamdan, "Optimization of neural network model using modified bat-inspired algorithm," Applied Soft Computing, vol. 37, pp. 71-86, 2015.

[5] P. Kora and S. Kalva, "Improved Bat algorithm for the detection of myocardial infarction," SpringerPlus, vol. 4, no. 1, p. 666, 2015.

[6] S. Roostaee and H. R. Ghaffary, "Diagnosis of Heart Disease Based on Meta Heuristic Algorithms and Clustering Methods," Journal of Electrical and Computer Engineering Innovations, vol. 4, no. 2, pp. 105-110, 2016.

[7] R. C. Deo, M. A. Ghorbani, S. Samadianfard, T. Maraseni, M. Bilgili, and M. Biazar, “Multi-layer perceptron hybrid model integrated with the firefly optimizer algorithm for windspeed prediction of target site using a limited set of neighboring reference station data,” Renew. Energy, vol. 116, pp. 309–323, 2018.

[8] A. Heidari, H. Faris, I. Aljarah, S. Mirjali, “An efficient hybrid multilayer perceptron neural network with grasshopper optimization,” Soft Comput, pp.1-18, 2018.

[9] Q. Han, H. Wu, T. Hu, and F. Chu, “Short-Term Wind Speed Forecasting Based on Signal Decomposing Algorithm and Hybrid Linear/Nonlinear Models,” Energies, vol. 11, no. 11, 2018

[10] Yang XS., “A New Metaheuristic Bat-Inspired Algorithm”, In: González J.R., Pelta D.A., Cruz C., Terrazas G., Krasnogor N. (eds) Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Studies in Computational Intelligence, vol 284. Springer, Berlin, Heidelberg

[11] N.S. Jaddi, S. Abdullah, A.R. Hamdan, “Optimization of neural network model using modified bat-inspired algorithm”, Appl. Soft. Comput. J. 37 (2015) 71–86.

[12] J. Yu, L. Xi, S. Wang, An improved particle swarm optimization for evolving feedforward artificial neuralnetworks, Neural. Process. Lett. 26 (3) (2007) 217–231.