Rainfall-Runoff modeling using Octonion-Valued Neural Networks

Shadab Shishegar*[1], Reza Ghorbani[2], Lyes Saad Saoud[3], Sophie Duchesne[1] and Geneviève Pelletier[4]

[1] Research Centre on Water, Earth, and the Environment, Institut National de La Recherche Scientifique (INRS), 490 Rue de La Couronne, Quebec City, QC, G1K 9A9, Canada.

[2] Department of Mechanical Engineering, University of Hawaii at Manoa, 2540 Dole St. Holmes Hall 201, Honolulu, HI, 96822, USA.

[3] Khalifa University, Al Saada St - Zone 1 - Abu Dhabi - United Arab Emirates

[4] Department of Civil Engineering and Water Engineering, Universit´e Laval, Pavillon Adrien Pouliot, 1065, Avenue de La M´edecine, Quebec City, QC, G1V 0A6, Canada

* Corresponding author: E-mail address: shadab.shishegar@ete.inrs.ca (S. Shishegar).

A B S T R A C T

Rainfall-runoff modeling is at the core of any hydrological forecasting system. High spatio-temporal variability of precipitation patterns, complexity of the physical processes, and large quantity of parameters to characterize a watershed make the prediction of runoff rates quite difficult. In this study, a hyper-complex Artificial Neural Network (ANN) in the form of an Octonion-Valued Neural Network (OVNN) is proposed to estimate runoff rates. Evaluation of the proposed model is performed using a rainfall time series from a rain gauge near a Canadian watershed. Results of the AI-generated runoff rates illustrate its capacity to produce more computationally efficient runoff rates when compared to those obtained using a physically-based model. In addition, training the data using the proposed OVNN versus a real-valued neural network shows less space-complexity (1*3*1 vs. 8*10*8, respectively) and more accurate results (0.10% vs. 0.95%, respectively), that accounts for the efficiency of the OVNN model for real-time control applications.

Keywords: Machine learning, Flow rate prediction, Stormwater management, Hydrology, Multi-dimensional, Hyper complex network

1- Introduction

The paradigm shift from using physically-based simulation to Artificial Intelligence (AI) in hydrological processes allows accurate modelling of complex systems without prior understanding of physical laws governing the process (Kalteh 2008). Artificial Neural Network (ANN) algorithms are among AI forecasting methods that have been widely employed in various hydrological fields particularly in the context of Climate Change (CC) (Daliakopoulos and Tsanis 2016) including rainfall-

runoff modelling (Kan et al. 2015; Tayyab 2019), flood prediction (Berkhahn et al. 2019) and long-term rainfall forecasting (Mekanik et al. 2013). All these hydrological phenomena are highly non-linear, time-varying and spatially distributed (Jurkar et al. 2009), and their mathematical representation requires detailed data on physical infrastructure, precipitation time series and hydrological characteristics of the studied watershed that could lead to high complexities in the modelling process. Rainfall-runoff simulation can be realized by either data-driven or physically-based models (Kan et al. 2015). Hernonin et al. (2013) report, in their state-of-the-art, that physically-based simulation models are not fast enough for real-time forecasting. However, the historical statistics and simple data-driven models have the ability to perform fast and reliably (Hernonin et al. 2013). Application of ANN as a data-driven model to estimate streamflow out of a rainfall data series was successfully employed during the last decades and the number of such applications has been growing fast (Daliakopoulos and Tsanis 2016). Even the popularity of ANN amongst hydrologists has been reported by American Society of Certified Engineering Technicians (ASCET) (Tayyab 2019).

The related literature shows that several procedures and architectures of ANN were proposed to deal with rainfall-runoff modelling. In Aytek et al. (2008), the performance of two ANN techniques, Feed Forward Back Propagation (FFBP) and Generalized Regression Neural Network (GRNN), was evaluated using historical hydro-meteorological data for the estimation of runoff in the Juniata River Basin (USA). Wavelet based ANNs is another type of neural networks that has been studied mostly to find a better correlation coefficient between the rainfall and runoff estimations (Dumka and Kumar (2021), Alizadeh, et al. (2021). Although this method can be employed for runoff forecasting, the large number of neurons in rainfall-runoff modelling studies may affect the correlation between the input and output features (Dumka and Kumar, 2021) to provide a reliable estimation of runoff. In another study by Mittal et al. (2012), a dual ANN was proposed to improve the performance of the flow prediction model in extreme events and compared to a Feed-Forward ANN (FF-ANN), which is widely present in the rainfall-runoff modelling literature. The developed dual ANN in that study outperformed the popular FF-ANN technique in prediction of high flows and it was suggested to be used under extreme events. Three other neural networks have been studied in Chen et al. (2014) where Copula-entropy theory was employed to skip the marginal and joint probability calculation in the ANN algorithm. Multi-layer FF-ANN, radial basis function networks and GRNN were chosen to evaluate the stream flow prediction performance of the system (Chen et al. 2014). In a recent study by Kao et al. (2020), an FF-ANN Encoder-Decoder algorithm is compared with a Long-Short Term Memory Encoder Decoder in order to provide an accurate flood forecasting for flood control systems. The proposed model in this study integrates a sequence-to-sequence learning procedure that converts the input sequence including hourly inflows, to the output sequence of reservoir inflow forecasts. As a result, they report that considering a sequence-to-sequence learning into the encoder-decoder algorithm can enable translation of the rainfall sequence into the runoff sequence while increasing the interpretability of the proposed model (Kao et al. 2020). Recent real domain ANN models have been developed to higher dimensional domains based

on which several hyper-complex techniques have been proposed. Among all these hyper-complex ANNs, Octonion-Valued Neural Networks (OVNNs) are proved to be one of the most promising approaches to model high-dimensional nonlinear processes (Saad Saoud and Ghorbani 2019). OVNNs consist of 8-dimensional inputs, outputs, weights and biases that are defined based on the Octonion numbers introduced by Conway and Smith (2003). In spite of Clifford algebras, the Octonion algebras are neither associative nor commutative i.e. $i_k i_l \neq i_l i_k \ \ \forall k \neq l$ and $i_k(i_l i_m) \neq (i_k i_l) i_m \ \ \forall k \neq l \neq m$.

This study is motivated by the need to model the complex process of rainfall transformation to runoff in real-time. As recommended in Shishegar et al. (2019), this complexity is inevitable specially in watershed-level investigations where the spatio-temporal variability of precipitation patterns is high, the associated physical processes are difficult to study and there are numerous parameters involved in the representation of the watershed. On the other hand, the OVNN algorithms are proved to be capable of providing efficient forecasting outputs for multi-dimensional complex problems relying on their two main features: 1) the ability to reduce the input-output dimensions by eight times; while 2) expanding the traditional backpropagation algorithm by adding seven other dimensions (Saad Saoud and Ghorbani 2019). Hence, in this study, an OVNN algorithm is developed as an alternative to physically based simulation models for runoff predictions in urban areas. To the best of our knowledge, the OVNN has not yet been employed in the literature for the estimation of runoff rates resulting from precipitation time series. Considering the real-time forecasting required in designing many *modern* urban stormwater management systems as a component of a greater whole named *smart city* (see e.g. Shishegar et al., 2020), the problem-solving speed is an important factor here.

In order to address the above problems, the scientific objectives of this study can be presented as follows:

- To propose an octonion-valued neural network algorithm to estimate the runoff rates at the outlets of a stormwater management network;
- To evaluate the performance of the proposed algorithms over the traditional physically based rainfall-runoff simulation approach in terms of not only the quality of prediction, but the computing time;
- To assess the outcomes of the multi-dimensional neural network for a real-case urban stormwater system; and
- To carry out a comparative analysis between the real-valued neural network and OVNN performances.

2- Experimental Area

The study area is an urban watershed located at the central province of Quebec, Canada. The catchment area has a surface of 311 ha of which 64% is residential, 14% is industrial, 9% is commercial and 13% is institutional, with an average impermeability of 62%. The physically based rainfall-runoff

simulation model of the sewer network of this catchment has already been developed and is used here for the comparative analysis of the proposed OVNN algorithm. PCSWMM software Version 7.0 was used for the rainfall/runoff simulation of the studied sector (Figure 1). This software performs based on the Stormwater Management Model - SWMM (Rossman and Huber 2016), that dynamically simulates stormwater runoff and flows in sewer networks from the specified rainfall series. In order to convert the current combined drainage network to a separated (stormwater) sewer network, all the wastewater flow values were given a zero value. This allows studying the runoff rates generated from the rainfall series which is here considered as the inputs of the proposed algorithm. Four storage units, designed at the outlets of the separate sewer network, are the points where the runoff rates will be estimated. A more detailed explanation of the input parameter structure is given in section 3-2.

Precipitation Data

An operating rain gauge located 80 km from the studied watershed measures the rainfall data by a tipping bucket and provides the observation record at a 5-minute time step. This recorded data is then validated by performing a comparative analysis with the recorded rainfall series by Environment Canada at the same station. In this study, the rainfall time-series of the year 2013 from May to November has been selected for the analysis of generated runoff. Generally, due to the meteorological characteristics of the Quebec Province region with long and snowy winters, the rainfall analysis is performed for the months of May to November. The data recorded for this period of the year 2013 shows a relatively higher average amount of rainfall (903 mm) compared to those recorded in average between 2000 and 2017 for the same period (759 mm). The reason to select this rainfall series for the investigations of this paper is to enable assessing the performance of the proposed OVNN algorithm under challenging conditions. Evidently, the more the proposed algorithm is trained based on critical data, the better its performance would be facing rainy periods. The rainfall characterization analysis is carried out based on two criteria for rainfall events separation: an inter-event duration of 6 hours and a minimum rainfall intensity of 1.2 mm/h. Table 1 shows the monthly and total rainfall height of the year 2013 in comparison to the average values (2000- 2017), along with the characteristics of this 2013 rainfall time series that provides a better understanding of the precipitation data used for ANN training. Also, the hyetograph of the 2013 rainfall series is illustrated in Figure 2.

3- Methodology

3-1- Octonion Valued Neural Network (OVNN)

This section develops the octonion valued neural network for training the rainfall and runoff data series in the form of octonion numbers given like:

$$O^{def} = x_1 + i_1x_2 + i_2x_3 + i_3x_4 + i_4x_5 + i_5x_6 + i_6x_7 + i_7x_8 \quad (1)$$

Where: $x_i, i \in \{1,2,\dots,8\}$ are the real parts and

$$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 \in \boldsymbol{R}$$

and $i_j, j \in \{1,2,\dots,7\}$ are the imaginary parts and

$$i_1^2 = i_2^2 = i_3^2 = i_4^2 = i_5^2 = i_6^2 = i_7^2 = -1$$

The latter includes three layers (Figure 3): an input layer with n octonion inputs, one hidden layer with m neurons, and one output layer with s neurons. These layers are related, respectively, to weights w_{nm}^1 and w_{ms}^2. The hidden and output layers have biases represented by w_{0m}^1 and w_{0s}^1, respectively. All network settings, inputs and outputs are considered octonion.

The j^{th} OVNN output can be calculated using the following equation:

$$\hat{y}_j(k+1) = f^2\left(\tilde{y}_j^{Re}\right) + \sum_{r=1}^{7} i_r f^2\left(\tilde{y}_j^{Im(i_r)}\right) \quad (2)$$

Where:

Re and $\mathrm{Im}(i_r)$ indices are the real and imaginary parts of $i_1, i_2, i_3, i_4, i_5, i_6,$ and $i_6,$ respectively.

f^2 is the sigmoid non-linear function given by the following equation:

$$f^2(.) = \frac{1}{1+e^{-(.)}} \quad (3)$$

$$\tilde{y} = \sum_{l=1}^{m} w_l^2 h_l + w_0^2 \quad (4)$$

Where:

$$l = 1, \dots, m$$

h_l is the l^{th} hidden neuron output, which is given by:

$$h_l = f^1\left(\tilde{h}_l^{Re}\right) + i_1 f^1\left(\tilde{h}_l^{Im(i_1)}\right) + i_2 f^1\left(\tilde{h}_l^{Im(i_2)}\right) + i_3 f^1\left(\tilde{h}_l^{Im(i_3)}\right) \quad (5)$$
$$+ i_4 f^1\left(\tilde{h}_l^{Im(i_4)}\right) + i_{15} f^1\left(\tilde{h}_l^{Im(i_5)}\right) + i_6 f^1\left(\tilde{h}_l^{Im(i_6)}\right) + i_7 f^1\left(\tilde{h}_l^{Im(i_7)}\right)$$

Where:

$$\tilde{h}_l = w_{nl}^1 u_n + w_{0l}^1 \quad (6)$$

u_n is the octonion valued vector of n octonion elements.

Noteworthy is that ReLU and sigmoid are the most employed non-linear activation functions in the literature for the hidden and output layers, both of which have the lower bound of zero. The sigmoid function transfers all the data to the bounded range between zero and 1, while ReLU keeps the upper bound of the data and converts all the negative values to zero. This feature of ReLU may be problematic as it decreases the ability of the ANN to train negative values by ignoring them. However, in rainfall-

runoff modeling where there is no negative data, the ReLU activation function can be used. In addition, ReLU is far more computationally efficient with a faster training process than the sigmoid function due to neurons with rectified functions that perform well to overcome saturation during the learning process as reported in Mboga et al. (2017). The sparsity and the reduced likelihood of vanishing the gradient are other advantages of ReLU that motivated us to use this activation function. The non-linear ReLU function is employed in the hidden layer which is given by the equation below:

$$f^1(x) = \max(0, x) \tag{7}$$

To optimize the network parameters, the octonion valued backpropagation is used. The objective is to optimize the parameters of the network in such a way that the total sum squared error in the output layer is minimized, which can be expressed as:

$$E = \frac{1}{2} e^c e = \frac{1}{2} \sum_d e_d e_d^* = \frac{1}{2} \sum_d E_d \tag{8}$$

$$E_d = e_d e_d^* = |e_d|^2 \tag{9}$$

Where:

The superscript '*' represents the conjugate operator;

C is the Cayley operator (Cayley 1846);

d is the number of samples; and

e^* is the error's conjugate.

The error *e* between the desired output y and estimated output $\hat{y}$ is:

$$e = y(k+1) - \hat{y}(k+1) = e^{Re} + \sum_{r=1}^{7} i_r e^{Im(i_r)} \tag{10}$$

In order to determine the optimal network parameters including weights and bias, the real valued delta rule proposed in Saad Saoud and Ghorbani (2019) is extended as follows:

The bias w_{0s}^2 is:

$$w_{0s}^2 = w_{0s}^{2Re} + i_1 w_{0s}^{2Im(i_1)} + i_2 w_{0s}^{2Im(i_2)} + i_3 w_{0s}^{2Im(i_3)} + i_4 w_{0s}^{2Im(4)} + i_5 w_{0s}^{2Im(i_5)} + i_6 w_{0s}^{2Im(i_6)} + i_7 w_{0s}^{2Im(i_7)} \tag{11}$$

We have:

$$\Delta w_{0s}^{2\ def} = w_{0s}^{2Re} + i_1 \Delta w_{0s}^{2Im(i_1)} + i_2 \Delta w_{0s}^{2Im(i_2)} + i_3 \Delta w_{0s}^{2Im(i_3)} + i_4 \Delta w_{0s}^{2Im(i_4)} + i_5 \Delta w_{0s}^{2Im(i_5)} + i_6 \Delta w_{0s}^{2Im(i_6)} + i_7 \Delta w_{0s}^{2Im(i_7)} = \Delta w_{0s}^{2Re} + \sum_{r=1}^{7} i_r \Delta w_{0s}^{2Im(i_r)}$$

$$= -\eta \left(\frac{\partial E}{\partial w_{0s}^{2Re}} + \sum_{r=1}^{7} i_r \frac{\partial E}{\partial w_{0s}^{2Im(i_r)}} \right) = -\eta \nabla_{w_{0s}^2} E$$

$$\rightarrow \nabla_{w_{0s}^2} E = \frac{\partial E}{\partial w_{0s}^{2Im(i_r)}} \quad (12)$$

$$\nabla_{w_{0s}^2} E = -\{e^{Re}(1-\hat{y}_s^{Re}).\hat{y}_s^{Re}\} + \sum_{r=1}^{7} i_r e_s^{Im(i_r)}\left(1-\hat{y}_s^{Im(i_r)}\right).\hat{y}_s^{Im(i_r)} \quad (13)$$

$$w_{0s}^2(k+1) = w_{0s}^2(k) - \eta \nabla_{w_{0s}^2} E \quad (14)$$

For the weights w_{ms}^2:

$$w_{ms}^2(k+1) = w_{ms}^2(k) - \eta \nabla_{w_{ms}^2} E \quad (15)$$

where

$$w_{ms}^2 = w_{ms}^{2Re} + \sum_{r=1}^{7} i_r w_{ms}^{2Im(i_r)}$$

and

$$\nabla_{w_{ms}^2} E = \frac{\partial E}{\partial w_{ms}^{2Re}} + \sum_{r=1}^{7} i_r \frac{\partial E}{\partial w_{ms}^{2Im(i_r)}} \quad (16)$$

$$\nabla_{w_{ms}^2} E = -h_{ms}^{*}.\left\{e^{Re}(1-\hat{y}^{Re}).\hat{y}^{Re} + \sum_{r=1}^{7} i_r e^{Im(i_r)}(1-\hat{y}^{Im(i_r)}).\hat{y}^{Im(i_r)}\right\} \quad (17)$$

For bias w_{0s}^1 and weights w_{nm}^1, the same procedure is used where:

$$w_{0m}^1 = w_{0m}^{1Re} + i_1 w_{0m}^{1Im(i_1)} + i_2 w_{0m}^{1Im(i_2)} + i_3 w_{0m}^{1Im(i_3)} + i_4 w_{0m}^{1Im(i_4)} + i_5 w_{0m}^{1Im(5)} + i_6 w_{0m}^{1Im(i_6)} + i_7 w_{0m}^{1Im(i_7)} \quad (18)$$

$$w_{nm}^1 = w_{nm}^{1Re} + i_1 w_{nm}^{1Im(i_1)} + i_2 w_{nm}^{1Im(i_2)} + i_3 w_{nm}^{1Im(i_3)} + i_4 w_{nm}^{1Im(i_4)} + i_5 w_{nm}^{1Im(5)} + i_6 w_{nm}^{1Im(i_6)} + i_7 w_{nm}^{1Im(i_7)} \quad (19)$$

The modification approach is therefore given as follows:

$$\nabla_{w_{0m}^2} E = \frac{\partial E}{\partial w_{0m}^{1Re}} + \sum_{r=1}^{7} i_r \frac{\partial E}{\partial w_{0m}^{1Im(i_r)}} \quad (20)$$

$$\nabla_{w_{0m}^2} E = \left\{(1-h_m^{Re}).h_m^{Re}.\nabla_{w_{0s}^2} E.w^{2*}(\nabla_{w_{0s}^2} E.w_{ms}^{2*})^{Re} + \sum_{r=1}^{7} i_r \left(1-h_m^{Im(i_r)}\right).h_m^{Im(i_r)}.(\nabla_{w_{0s}^2} E.w_{ms}^{2*})^{Im(i_r)}\right\} \quad (21)$$

$$w_{0m}^1(k+1) = w_{0m}^1(k) = -\eta \nabla_{w_{0m}^1} E \quad (22)$$

$$\nabla_{w_{nm}^1} E = \frac{\partial E}{\partial w_{nm}^{1Re}} + \sum_{r=1}^{7} i_r \frac{\partial E}{\partial w_{nm}^{1Im(i_r)}} \quad (23)$$

$$\nabla_{w_{nm}^1} E = -u_n^{*}.\nabla_{w_{nm}^1} E \quad (24)$$

$$w_{nm}^1(k+1) = w_{nm}^1(k) = -\eta \nabla_{w_{nm}^1} E \quad (25)$$

With η as the learning rate.

Note that the conjugate of an octonion number is:

$$o^{*def} = x_1 - i_1 x_2 - i_2 x_3 - i_3 x_4 - i_4 x_5 - i_5 x_6 - i_6 x_7 - i_7 x_8 \quad (26)$$

3-2- Determination of Input Structure

Following the discussion provided in the « study area and database » section, the input data can be computed by running the PCSWMM simulation model in order to generate, from the rainfall series,

the inflow rates at the four outlets of the studied drainage network.

The OVNN model is built based on the combination of the recorded rainfall time series and runoff outflows simulated by PCSWMM at the four outlets of the network. The rainfall measures in eight consecutive time-steps is combined to create one single input parameter for the feed-forward network. In total, 61,630 5-minute records are available over the period of May-November 2013 for the studied area. This data set is divided into two subsets for training and testing, as shown in Table 2, where the parameter $data_{min}$ represents the minimum value, $data_{max}$ is the maximum value, S_{data} is the standard deviation, and $\overline{data}$ is the mean for each subset, separately. As for the proposed methodology, the first 85% of the data is employed for training of the OVNN with 15% of the remaining data for testing based on which all the performance criteria are calculated.

As seen in Table 1, the most critical months in terms of rainfall height are May and August with reported total rainfall of 180 mm and 166 mm, respectively. However, the highest daily rainfall heights are recorded in August and June with 73 mm and 56 mm, respectively. Also, looking at the data recorded over other years, like the year 2007 for instance, some extreme events occurred in May, September and October with no reported extreme event in August and June. This shows that the occurrence of extreme events can be in any month and also highlights the importance of the input data used to train the model and how these data may sometimes be noisy, correlated or even with no relevance to the output parameters (Chen et al. 2014). In this study, the data is modified in such a way that it becomes feedable to the OVNN-forecaster. For this purpose, all the data are defined in the octonion domain as shown in Figure 3. This data preparation allows employing the eight rainfall height predicted over the next eight 5-minute time-steps (40 minutes in total) for prediction of upcoming runoff rates during a 40-minutes period which starts from 10 minutes in the future due to the runoff delays. It means that each 8 time-step of rainfall data (x_i) creates a single unit ($u_i(k)$) to feed to the neural network to finally predict the output units ($\hat{y}_i(k)$) each of which consists of 8 consecutive runoff flow rates ($\hat{o}_i$) (Figure 3).

As the last step of the methodology, the available dataset related to the year 2013 is used to train a real-valued neural network (RVNN) in order to evaluate the advantages of using the OVNN-forecaster. The designed RVNN algorithm is a Multi-Layer Perceptron (MLP) algorithm which consists of 8 input/output pairs. The MLP neural networks are popular ANNs n time-series forecasting studies that represent promising target values estimations (Ahmad, et al., 2010). 61,630 samples, 85% of which is used for training with 15% for testing. Ten neurons over one hidden layer are considered resulting a total of 178 parameters in the network including weights and biases. Besides, coherent with the proposed OVNN, the ReLU activation function is employed for non-linearity of the outputs from the neurons in RVNN.

4- Results and Discussion

Figure 4 illustrates the hydrographs of the simulated (with the SWMM model) versus predicted

(with the OVNN model) flow rates at the outlets of the studied watershed. As can be seen from Figure 3, the OVNN model is able to reasonably estimate the flow rates out of the precipitation data of the year 2013. To provide a quantitative analysis of the OVNN-forecast accuracy, two performance criteria are considered: 1) the Normalized Root Mean Squared Error (nRMSE), and 2) the Mean Absolute Error (MAE). nRMSE is employed here to facilitate the comparison of model performance for different stormwater outlets that may have different flow rate scales. Also, MAE is a common metrics in neural network studies with the ability to measure the accuracy for continuous variables. Table 3 compares the value of these two performance criteria for each stormwater outlet separately. It can be seen that the runoff estimation by OVNN is carried out with small nRMSE and MAE, respectively less than 4% and 0.2%, when compared to the SWMM model outputs. Besides, the sequence-to-sequence process of the proposed OVNN that converts the rainfall input sequence to the runoff output sequence (Kao et al., 2021) includes 61,737 training samples. This indicates that the training process of OVNN from scratch with a space complexity of 1*3*1, reduces the number of samples by eight times in the input layer compared to the RVNN approach with the space complexity of 8*10*8 and takes an average time of 2 hours and 31 minutes using a PC computer Core i7 16GB GPU for extracting the data related to a 48-hour period. Once it is trained, the prediction can be performed almost instantly. While the simulation using SWMM for the same period takes more than 26 minutes. Noteworthy is that the training of the OVNN algorithm should be frequently repeated as a background process, in order to keep the performance of the algorithm updated regarding the new arriving data. However, predicting the outflow rates can be done in some seconds to enable keeping up with the real-time control process.

Figure 5 shows the linear fit between the values simulated by the SWMM and OVNN models, for all studied outlets, to evaluate the performance of the OVNN forecaster. As a result of this univariate linear analysis, obtained using the least-squares fit polynomial method, the coefficient of correlation (R) along with the forecast intercepts are calculated to determine the goodness of fitness of the model forecaster as shown in Table 4.

As reported in Table 4, the high values of R ($\cong$ 1) and low values of b ($\cong$ 0) show the ability of the proposed model to determine the runoff data however, for higher flow rates, more variations from the simulated values are shown. Here, since more than 96% of the sample data are related to flow rates less than 1.3 m^3/s , the regression line does not represent well the higher rates especially in outlets 1 and 2. This implies the importance of feeding validated input data to the model forecaster in order for it to properly estimate the predictions. As aforementioned, since the model training based on more critical meteorological conditions is more beneficial, the rainiest year (the year 2013) was selected in this study to train the OVNN.

As a further analysis and in order to validate the advantages of the introduced OVNN, the data of the studied case was also used to train a real-valued neural network (RVNN). To this purpose, the

RVNN algorithm was developed with varying number of iterations from 200 to 20000 to ensure the errors have already been reduced. The results from the performance criteria obtained using RVNN modelling approach are shown in Table 5.

In Table 6, the comparative analysis between the performances of the introduced OVNN with those of the real-valued neural network for a one week period shows that the developed RVNN performs slower in comparison to the OVNN while the accuracy of OVNN in estimation of runoff rates is much higher. The implementation of RVNN can achieve the same level of accuracy only by allocating more iterations, which causes a significant increase in running time. On the other hand, the total size of the input vector for the RVNN is eight times more than the input vector of OVNN. Furthermore, the space complexity of OVNN is significantly less than the RVNN with a size of 1*3*1 (excluding bias) versus 8*10*8, respectively. Although the OVNN architecture can be extended to support longer term predictions, a neural network with more neurons in the output layer has to endure a relatively higher training and prediction time, which makes it inappropriate to use in real-time control applications.

5- Conclusion

Physically based models have been employed for several years for rainfall-runoff modelling, however advances in technology along with the emergence of real-time control systems, created a need to employ faster tools to generate real-time forecasting data. Octonion-valued neural networks as a multi-dimensional network was introduced in this paper for representing complex problems like rainfall-runoff hydrological modeling. Through this study, precipitation time series data is used to model the flow rate data based on an octonion neural network algorithm, where the ReLU is employed as the activation function. Simulated flow rates using the physically-based simulation model were used to train the proposed OVNN-forecaster and, furtherly, the performance of the proposed model in estimating runoff flow rates was compared with those obtained using a real-valued neural network. Results showed that using the OVNN model was beneficial in terms of run time and accuracy making it an efficient, fast and reliable tool for decision makers in real-time controllers that finally serve as a small, yet effective, component of a greater whole named *smart city*.

It was also shown that the model is less accurate for more intense rain events. Hence, it would be beneficial to train the model based on more critical meteorological conditions. In addition, a metacognitive component can be added to the OVNN-forecaster to enable self-regulation of the network parameters during the learning process. This capability provides an enhanced learning process by selecting more critical samples to train the model. In all cases, keeping the computational efficiency of the algorithm to achieve smaller running times yet more accurate predictions, should be taken as the highest priority.

REFERENCES

Alizadeh, A., Rajabi, A., Shabanlou, S., Yaghoubi, B., & Yosefvand, F. 2021. Modeling long-term rainfall-runoff time series through wavelet-weighted regularization extreme learning machine. Earth Science Informatics, 1-17.

Aytek, A., Asce, M., and Alp, M. 2008. An application of artificial intelligence for rainfall – runoff modeling. Journal of Earth System Science, 117(2), 145-155.

Berkhahn, S., Fuchs, L., and Neuweiler, I. 2019. An ensemble neural network model for real-time prediction of urban floods. J. Hydrol. 575 (February): 743–754. Elsevier. doi:10.1016/j.jhydrol.2019.05.066.

Cayley, A. 1846. On the rotation of a solid round a fixed points. Cambridge Dublin Math. J. 2(1): 167–183.

Chen, L., Singh, V.P., and Guo, S. 2014. Copula entropy coupled with artificial neural network for rainfall – runoff simulation. Stochastic environmental research and risk assessment, 28(7), 1755-1767. doi:10.1007/s00477-013-0838-3.

Conway, J.H., and Smith, D.A. 2003. On Quaternions and Octonions. Taylor & Francis, New York.

Daliakopoulos, I.N., and Tsanis, I.K. 2016. Comparison of an artificial neural network and a conceptual rainfall – runoff model in the simulation of ephemeral streamflow. Hydrol. Sci. J. 61(15): 2763–2774. Taylor & Francis. doi:10.1080/02626667.2016.1154151.

Dumka, B.B., and Kumar, P. Modeling rainfall-runoff using artificial neural network (ANNs) and wavelet based anns (WANNs) for Haripura Dam, Uttarakhand. Indian Journal of Ecology 48.1 (2021): 271-274.

Environment Canada. 2019. Historical precipitation data. Available from https://climate.weather.gc.ca/historical_data/search_historic_data_e.html. 2020-06-11.

Hernonin, J., Russo, B., Mark, O., and Gourbesville, P. 2013. Real-time urban flood forecasting and modelling – a state of the art. Journal of Hydroinformatics, 15(3), 717-736. doi:10.2166/hydro.2013.132.

Jurkar, M.P.R.A., Kothyari, U.C., and Chaube, U.C. 2009. Artificial neural networks for daily rainfall — runoff modelling. Hydrological Sciences Journal, 47(6), 865-877. doi:10.1080/02626660209492996.

Kalteh, A.M. 2008. Rainfall-Runoff Modeling Using Artificial Neural Networks (ANNS): Modelling and Understanding. Caspian Journal Of Environmental Sciences (CJES) 6(1): 53–58.

Kan, G., Yao, C., and Li, Q. 2015. Improving event-based rainfall-runoff simulation using an ensemble artificial neural network based hybrid data-driven model. Stoch. Environ. Res. Risk Assess.: 29(5), 1345-1370. Springer Berlin Heidelberg. doi:10.1007/s00477-015-1040-6.

Kao, I-Feng, Zhou, Y., Chang, L. C., and Chang, F. J. 2020. Exploring a Long Short-Term Memory

based Encoder-Decoder framework for multi-step-ahead flood forecasting. Journal of Hydrology 583, 124631.

Mboga, N., Persello, C., Bergado, J. R., & Stein, A. 2017. Detection of informal settlements from VHR images using convolutional neural networks. Remote sensing, 9(11), 1106. doi.org/10.3390/rs9111106

Mekanik, F., Imteaz, M.A., Gato-trinidad, S., and Elmahdi, A. 2013. Multiple regression and Artificial Neural Network for long-term rainfall forecasting using large scale climate modes. J. Hydrol. 503: 11–21. Elsevier B.V. doi:10.1016/j.jhydrol.2013.08.035.

Mittal, P., Chowdhury, S., Roy, S., Bhatia, N., and Srivastav, R. 2012. Dual Artificial Neural Network for Rainfall-Runoff Forecasting. Journal of Water Resource and Protection, 4(12), 1024-1028.

Rossman, L.A., and Huber, W.C. 2016. Storm Water Management Model Reference Manual Volume I – Hydrology. U.S. Environmental Protection Agency. Environmental Protection Agency, Cincinnati, OH, USA. Available from www2.epa.gov/water-research.

Saad Saoud, L., and Ghorbani, R. 2019. Metacognitive Octonion-Valued Neural Networks as They Relate to Time Series Analysis. IEEE Transactions on Neural Networks and Learning Systems 31 (2), 539-548

Shishegar, S., Duchesne, S., and Pelletier, G. 2019. An integrated optimization and rule-based approach for predictive real time control of urban stormwater management systems. J. Hydrol. 577(June): 124000. Elsevier. doi:10.1016/j.jhydrol.2019.124000.

Shishegar, S., Duchesne, S., Pelletier, G., and Ghorbani, R. 2021. A smart predictive framework for system-level stormwater management optimization. Journal of Environmental Management. 278: 111505. ISSN 0301-4797. doi.org/10.1016/j.jenvman.2020.111505.

Tayyab, M. 2019. Rainfall-runoff modeling at Jinsha River basin by integrated neural network with discrete wavelet transform. Meteorol. Atmos. Phys. 131(1): 115–125. Springer Vienna. doi:10.1007/s00703-017-0546-5.

Tables:

Table 1- Characteristics of 2013 rainfall series with an inter-event duration of 6 h and comparison to the total rainfall height of 2000-2017 at the considered station (Environment Canada 2019)

2013 Rainfall Characteristics									
Month	May	Jun	Jul.	Aug.	Sep.	Oct.	Nov.	Total	Avg. 2000-2017
Total rain depth (mm)	180.8	154.5	70.0	166.9	145.8	100.2	85.7	903.9	759.5

Number of events	74
Average water height/event (mm)	8.94
Average intensity/event (mm/h)	1.84
Average maximum intensity over 10 min (mm/h)	12.08
Average duration (h)	6.23

Table 2- Characteristics of training and testing data

	Training set	**Testing set**
$data_{min}$ (mm)	0	0
$data_{max}$ (mm)	73.14	35.88
S_{data} (mm)	1.455	1.008
$\overline{data}$ (mm)	0.1878	0.1554

Table 3- OVNN performance criteria calculations for the four studied outlets

	nRMSE (%)	**MAE (%)**	**Max. Simulation**	**Max. Prediction**
Outlet 1	2.8705	0.0817	4.29 m^3/s	3.98 m^3/s
Outlet 2	3.4481	0.1206	6.47 m^3/s	6.87 m^3/s
Outlet 3	3.8263	0.1458	7.39 m^3/s	6.01 m^3/s
Outlet 4	2.3319	0.0549	7.01 m^3/s	6.82 m^3/s

Table 4- The regression analysis parameters calculated for the four studied outlets

	Coefficient of correlation (R)	Forecast intercept (b)
Outlet 1	0.8601	0.0107
Outlet 2	0.9549	0.0081
Outlet 3	0.9118	0.0103
Outlet 4	0.9689	0.0234

Table 5- RVNN performance criteria calculation with considering 20000 and 200 iterations

Performance Criteria	nRMSE (%)	MAE (%)	nRMSE (%)	MAE (%)
Number of Iterations	**20000**		**200**	
Outlet 1	10.52	1.061	10.71	1.032
Outlet 2	9.93	1.053	10.26	1.002
Outlet 3	9.75	0.929	9.89	0.973
Outlet 4	9.16	0.901	9.35	0.925

Table 6- Comparison between the RVNN and OVNN models over the four studied outlets for a one-week period

Network	Iterations	Parameters	Neural Network Architecture	Average nRMSE (%)	Average MAE (%)	Training Time (sec.)
RVNN	20000	178	8x10x8	9.84	0.986	10523
OVNN	2000	10	1x3x1	3.05	0.173	115

Figures:

Figure 1- Simulation model of the studied sector using SWMM

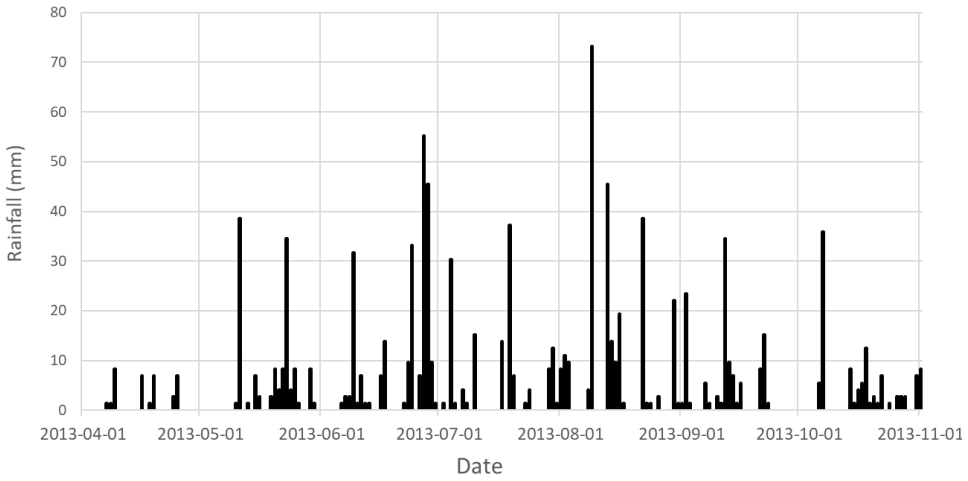

Figure 2- Time series plot of 2013 daily rainfall data for the period May to November.

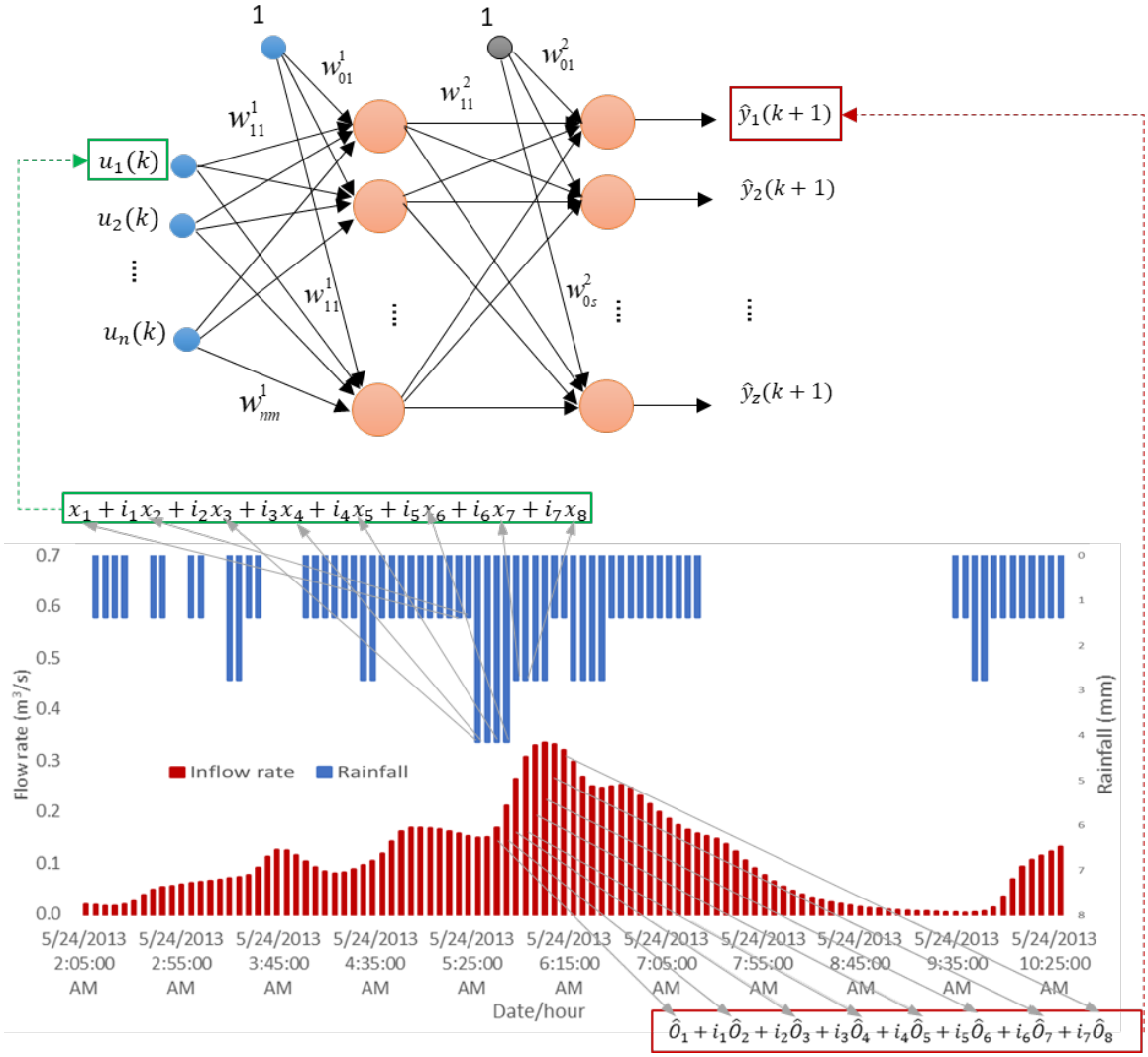

Figure 3-Octonion valued neural network architecture and its associated rainfall-runoff parameters

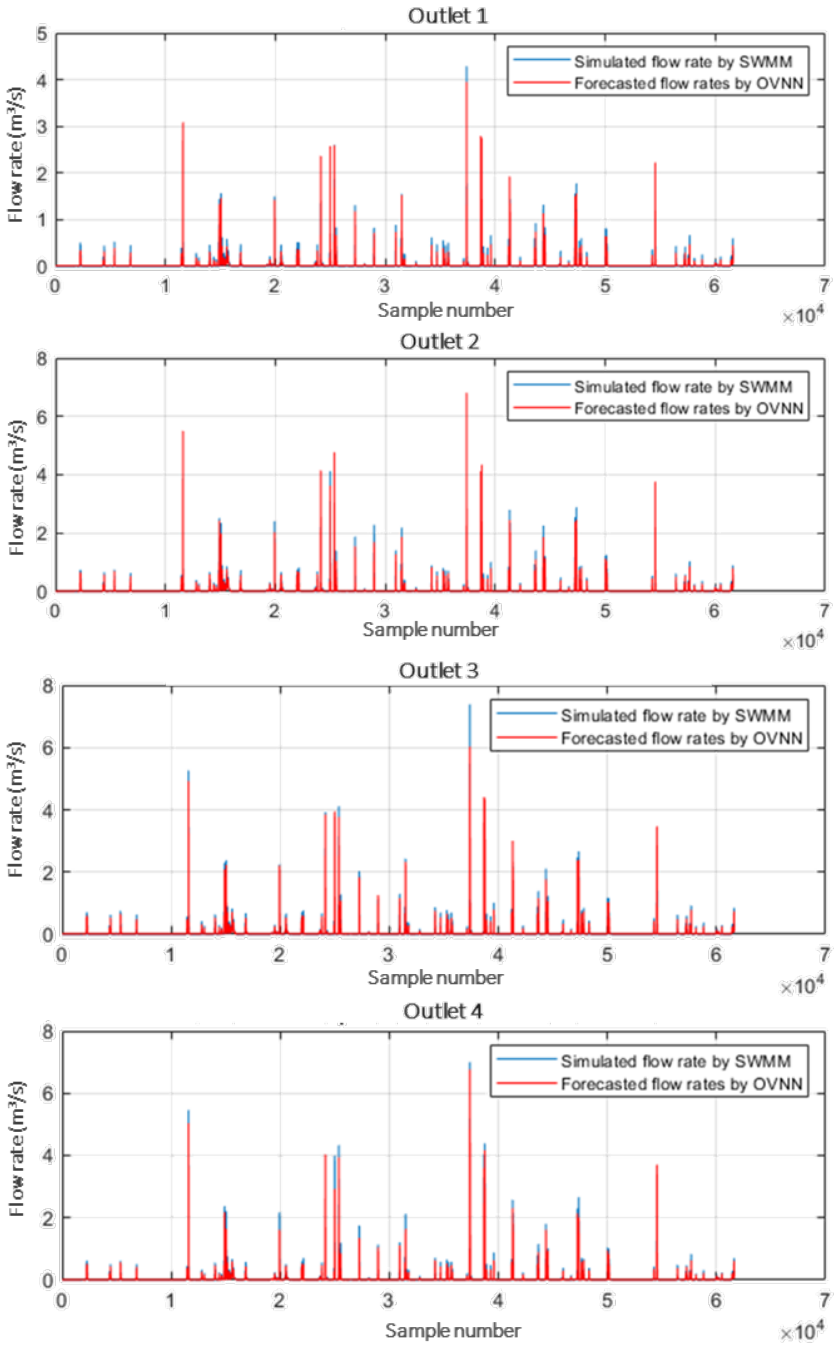

Figure 4-Simulated flow rates by SWMM versus forecasted flow rates by OVNN for the data samples of the year 2013

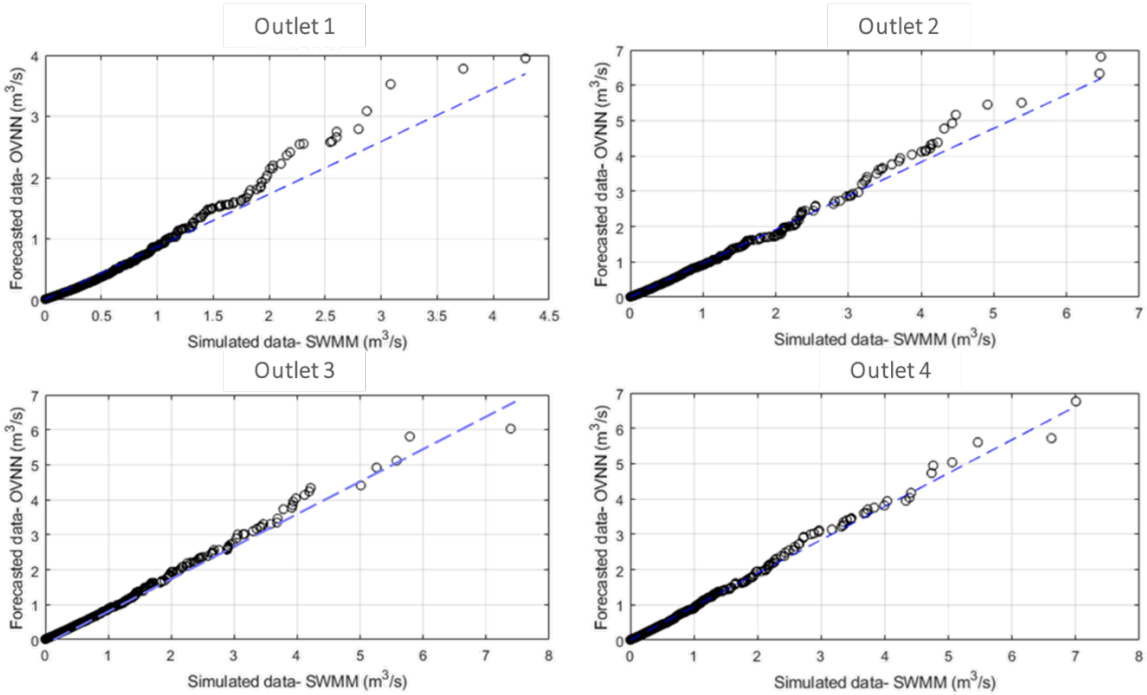

Figure 5- Univariate linear regression analysis of the simulated and estimated flow rates for the four studied stormwater outlets

List of Figures

List of Tables