Water Leakage Detection using Neural Networks

Shreya Sabu
North Carolina State University
ssabu2@ncsu.edu

Gnanamanikam Mahinthakumar
North Carolina State University
gmkumar@ncsu.edu

Ranji Ranjithan
North Carolina State University
ranji@ncsu.edu

James Levis
North Carolina State University
jwlevis@ncsu.edu

Downey Brill
North Carolina State University
brill@ncsu.edu

Abstract: The primary goal of the project is to leverage recent developments in smart water technologies to detect and reduce water leakages in large water distribution networks with the aid of neural networks. A cost effective non-invasive solution to detect leakages in transmission pipelines is needed by many water utilities as it will lead to significant water savings and reduced pipe breakage frequencies, especially in older infrastructure systems. The eventual goal of the project is to test the ANN model on a real network using field measured pressure and pipe breakage data after tuning and developing the model with simulated data. In this project we propose building a regression model, based on Multi-Layer Perceptron (MLP) algorithm, which is a class of feedforward Artificial Neural Networks (ANNs) to detect the leak locations within a proposed network. The model should be able to learn the structure, i.e. mapping of various leak nodes and sensor nodes in an area, such that it can detect the leak nodes based on the pressure values with significant accuracy.

I. INTRODUCTION

Recent advances in Cloud Computing technologies, the Internet of Things (IoT), and smart water technologies make it possible to develop automated systems that can generate real-time reports of leakage locations in a water distribution system using routinely collected data. The primary goal of this research is to develop machine learning techniques that can leverage real-time data, hydraulic modeling, and Cloud technologies to detect and reduce water leakages in water distribution systems. The leakage detection analytics module is implemented using a Multi-Layer Perceptron Algorithm (MLPA) based on artificial neural networks. The model will learn the structure, i.e., mapping of various leak nodes with pressure responses at sensor nodes in an area with an aid of a hydraulic model, to detect leak nodes based on the pressure readings with reasonable accuracy. The MLPA model is trained, validated, and tested using pressure data generated with simulated leaks using a hydraulic model for different test networks with various levels of model and measurement errors. Data generation and training are performed on local computing resources as well as resources available on the google cloud platform. The efficacy of the algorithm is tested for both regression (size of leaks) and classification (whether or not a leak is present) accuracy. Preliminary results for a medium-sized network show that classification accuracy exceeds 90 percent even in the presence of significant noise. This can lead to a cost-effective solution to detect leakages in water transmission lines from routine pressure measurements, which in turn leads to significant water savings and reduced pipe breakage frequencies, especially in older systems. Consumers will see lower water costs and reduced disruptions to services as more and more utilities adopt such technologies.

Data Generation: The data is in the form of leak values and pressure values simulated using the WNTR software (https://wntr.readthedocs.io/en/latest/). WNTR is used to model the water distribution systems. An illustrative water distribution network with candidate leak nodes and pressure sensor nodes is shown in Fig.1.

The simulated dataset will contain leak values and pressure responses for any random number of leaks with random leak sizes (Fig.2). This is done for 100,000 realizations. Similarly, the pressure response values represent the pressure differences at the sensor nodes due to the leaks present.

Fig. 1. Illustrative Water Distribution Network

Multi-Layer Perceptron (MLP): A multi-layer perceptron (MLP) is a deep, artificial neural network. They are composed of an input layer to receive the signal, an output layer that makes a decision or prediction about the input,

Sample Leak Values															
Node1	Node2	Node3	Node4	Node5	Node6	Node7	Node8	Node9	Node10	Node11	Node12	Node13	Node14	Node15	Node16
0	2	0	0	0	0	0	3	0	0	0	0	4	0	0	0
0	4	0	0	0	0	0	0	0	8	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0
0	8	0	0	0	6	0	3	10	0	0	5	0	0	0	0

Sample Pressure Values															
Node1	Node2	Node3	Node4	Node5	Node6	Node7	Node8	Node9	Node10	Node11	Node12	Node13	Node14	Node15	Node16
0.027035	0.051872	0.067543	0.039513	0.031929	0.012375	0.028687	0.043064	0.052933	0.058994	0.07378	0.107937	0.152534	0.101807	0.091793	0.056034
0.045982	0.089344	0.117271	0.06694	0.054207	0.021233	0.048775	0.072491	0.087807	0.092964	0.133488	0.087234	0.080238	0.070751	0.064766	0.091915
0.011093	0.016731	0.02047	0.016373	0.013245	0.004726	0.01181	0.018921	0.025459	0.027477	0.051662	0.130043	0.144535	0.207073	0.14444	0.028271
0.122879	0.233437	0.317451	0.178909	0.144608	0.0564	0.130341	0.194996	0.224495	0.236866	0.224968	0.222637	0.264748	0.20052	0.182701	0.213291

Fig. 2. Sample Leak and Pressure Values

and in between those two, an arbitrary number of hidden layers that are the true computational engine of the MLP. MLPs with one hidden layer are capable of approximating any continuous function. Multilayer perceptrons are often applied to supervised learning problems. They train on a set of input-output pairs and learn to model the correlation (or dependencies) between those inputs and outputs. MLPs are known for two main operations: In the forward pass, the signal flow moves from the input layer through the hidden layers to the output layer, and the decision of the output layer is measured against the ground truth values. In the backward pass, using backpropagation and the chain rule of calculus, partial derivatives of the error function with respect to the various weights and biases are back-propagated through the MLP. A multi-layer perceptron model is developed which helps to identify the leak nodes based on the pressure values which are given as an input.

A regression model based on Multi-Layer Perceptron algorithm is built, which will predict the regressed continuous leak values of all the leak nodes based on their respective pressure values. The node is classified as a leak if the predicted leak value crosses a certain threshold.

Related Work: The paper [1] describes a model based on Self Organising Maps(SOMs) and Multi Layer Perceptron Algorithm (MLP) for detection and localisation of water leaks. The models based on SOMs and MLPs for the task of detection of leaks, achieved an accuracy of 73% and 85% respectively.

The detection of the presence of leaks in pipeline transport systems using Multilayer Perceptron is discussed in the paper [2]. A probabilistic model was built which correlates the measurements of inlet and outlet pressures and also the flow to the state of leakages. The paper suggests that MLP provides good performance in leak detection with fast processing and sorting capability.
The toolbox and libraries that we have used for this project is shown Table 1.

Libraries	Functions
sklearn	Data preprocessing, accuracy measure, predicting probabilites, confusion matrix etc
wntr	Simulate pressure for the given water network model
matplotlib	Plotting learning and complexity curves
pandas	For reading the data files from csv
keras	Load different model layers, tweak hyperparamters, predict the test data using the model.

Table 1.

A. Model

Implemented a regression model based on Multi Layer Perceptron Algorithm. Model follows a general network architecture pipeline which is given in Fig 3.

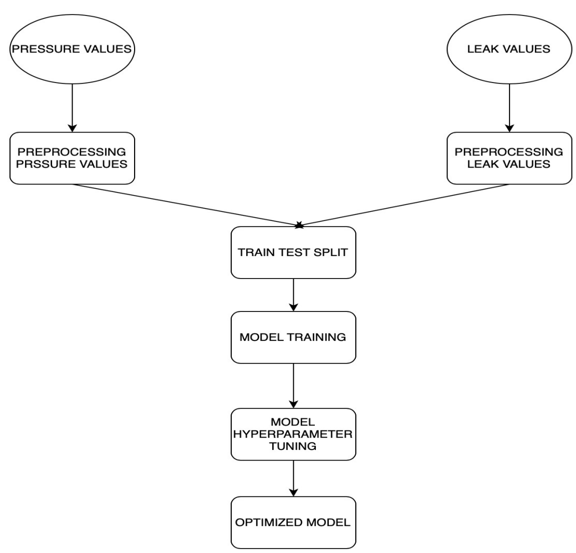

Fig. 3. Network Architecture

Multi-Layer Perceptron(MLP) Model: Implementation of a regression model is based on MLP algorithm. The main goal of the model was to detect the leak nodes based on the pressure values. Multilayer perceptron is a supervised type of learning model. The dataset is split into 80% training set, which is further divided in the ratio 80:20 for training set and validation set, and 20% testing set and the network is built over a sequential model.

The model consists of pressure nodes as input and leak nodes as output. The hidden layers form the dense layers. These layers are fully connected layers, that is, all the neurons in a layer are connected to those in the next layer. Densely connected layer provides learning features from all the combinations of the features of the previous layer. The model consists of 3 dense layers as part of the hidden layers. There are 16 neurons in each of these hidden layer. The MLP network model is as shown in Fig. 4.

The model is further trained using the training data and classified using the testing data. The model uses the Nadam optimizer and Mean Absolute Error (MAE) as the metrics. MAE gives the average over the test samples of the absolute differences between prediction and actual observation values where all individual differences have equal weight. Along with accuracy, the precision and recall values for each node was also calculated. This is done in order to minimize the false negatives which is used to predict a no leak when it is actually a leak. Accuracy gives the total number of correct predictions out of all the observations. Precision is the number of correct predictions for null hypothesis i.e. the leaks. Recall suggests the total number of leak observations explained by the model. Table 2. gives a brief description of the values of predicted leak for actual leak and no leak present at a particular node. From these values, the precision, recall and accuracy of the model is calculated.

	Predicted Leak	Predicted No Leak	Total
Actual Leak	5832	136	5968
Actual No Leak	2828	23204	26032
Total	8660	23340	32000

Table 2.

II. EXPERIMENTAL SECTION

A. Metrics

The main objective is to classify whether a node is a leak node or not based on leak node data and pressure values. Different regression models are evaluated and compared, which are based on Multi-Layer Perceptron, based on mean absolute error, recall and accuracy.

B. Model Selection

Multi-Layer Perceptron Model: The hyper parameters which were considered to implement the MLP model are given as follows:

- Loss Function
- Activation Function
- Number of Epochs
- Batch Size

As discussed earlier, the MLP model has 3 hidden dense layers and 16 pressure nodes as inputs and 16 leak nodes as outputs. Experimented with different number of hidden layers and number of neurons in those layers as well. Initial model created had only 1 hidden layer and 32 units in that layer. As the number of hidden layers increased, the accuracy went up and mean absolute error went down, until 3 hidden layers, after that the model accuracy went down and mean absolute error went up implying overfitting. Similar pattern was observed when experimented with number of neurons in a hidden layer, the accuracy went down and mean absolute error increased when the number of neurons increases more than 16. Hence, the final model has 3 hidden layers and 16 neurons in each. Experimentation was also done on the activation used. Initially, ReLU was used as an activation function in the model. ReLU units may get fragile during training and may die. A large gradient flowing through a ReLU neuron could cause the weights to update in such a way that the neuron will never activate on any datapoint again. ReLU units can irreversibly die during training since they can get knocked off the data manifold. Thus, the neurons which never activate in the training dataset may result into some part of the network to be dead. This was observed while training the model and out of the 16 nodes, 3 nodes showed no leak at a given time. This is called as the 'dying ReLU' problem and can be shown using the confusion matrix for node 16, which suggests that the predicted leak value will always be zero.

Thus, instead of using ReLU function, Leaky ReLU was used as an activation function in the final model. Though the accuracy of the model was low when compared with the model using ReLU, recall was improved which is an important parameter to be considered in order to detect the leak nodes correctly. Predicting no leak when actually a leak is present for the given node may lead to undesirable results.

Sigmoid activation function was used as another variation of hyperparameter which exists in the range of [0,1]. The pressure values and leak values were standardized in order to scale the values between (0,1) when using the sigmoid function. The accuracy obtained after using sigmoid function was 73.28%, which was very low when compared with other activation functions such as ReLU and Leaky ReLU. In addition to this, the Mean Absolute Error (MAE) obtained was 0.305. The MAE value for Leaky ReLU was 0.2, which is far lower than that obtained using sigmoid function. This was carried out for 1000 epochs. The model was further trained by increasing the epochs to 5000 with sigmoid activation function. The accuracy for this model was observed to be 82.66%. The lower accuracy using sigmoid function may be because of scaling the high leak values in the range [0,1]. This in turn affected the classification of the leaky node. Thus, the Leaky ReLU outperformed other activation functions considered which was used in the final model. The other parameter considered for hyperparameter tuning was the threshold value. It was set to zero which suggested that when the predicted value was greater than zero, the node was considered to be a leak node.

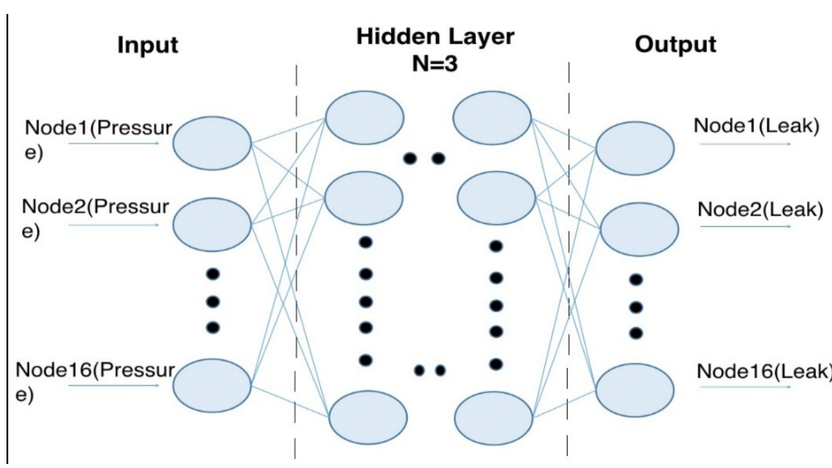

Fig. 4. Multilayer Perceptron Model

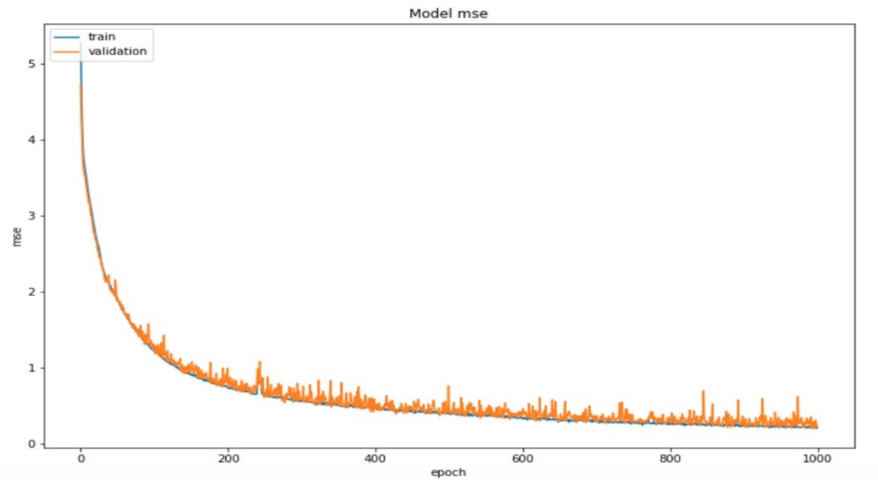

Fig. 5. Model Loss of Final Model

Activation Function	Epochs	Batch Size	Accuracy	MAE
ReLU	3000	8	93%	0.24
Leaky ReLU	1000	8	97.23%	0.2
Sigmoid	1000	16	73.28%	0.305
Sigmoid	5000	8	82.66%	0.25
Sigmoid	5000	32	79.83%	0.28

Table 3. Hyperparameter tuning

	Predicted Leak	Predicted No Leak
Actual Leak	0	205
Actual No Leak	0	1795

Table 4.

Threshold	Accuracy
0.1	79.26%
0.4	96.15%
0.8	97.13%
0.9	97.23%

Table 5.

This threshold was later varied from 0.1 to 0.9 in steps of 0.1 and the accuracy varied accordingly as shown in Table 5. It is observed that the optimal threshold value was 0.9 as it had the highest accuracy of 97.23%. Also, the leak sizes considered in the model range from 1 to 5. It can be inferred that leak size with value 1, which is the smallest leak node, is the only leak size which has a probability of misclassification for the threshold value 0.9. This is clear from the below given table.

The hyper parameter tuning is performed by changing parameters as discussed above considered for the model. The results for each simulation are briefly given in the table below. The highest accuracy after comparing these models was 97.23%. The graph shown in in the Fig. 5 clearly shows that the mean absolute error is decreasing as the total number of epochs were increasing.

C. Performance and Comparison to the Baseline

Model	**Accuracy**
MLP 10-14-1 (Base)	85.4%
MLP Model (Proposed)	97.23%

Table 6

It can be observed in Table 6 that the MLP based model clearly outperforms the base model.

III. CONCLUSION

The water pipes network was built using the EPANET/WNTR software. Total 16 number of pressure and leak nodes were considered and the network was simulated from which the pressure and leak values respectively for each node was used as input to the model. All the experimentation was done with a pre-determined number of leak nodes and pressure nodes. Further experimentation with different combinations and geographic placements of both leak and pressure nodes can be experimented. Also, experimentation with models based on other deep learning techniques could be done to check whether there can be an improvement in the current model for better predictions and localization of leaks in smart water management systems.

Acknowledgements: This material is based upon work partially supported by the National Science Foundation under Partnership for Innovation (PFI) grant no. 1919228.

REFERENCES

[1] I. Rojek and J. Studzinski, "Detection and localization of water leaks in water nets supported by an ict system with artificial intelligence methods as a way forward for smart cities," *Sustainability*, vol. 11, no. 2, p. 518, Jan 2019. [Online]. Available: http://dx.doi.org/10.3390/su11020518

[2] J. Gomez-Camperos, E. Espinel-Blanco, and F. Regino-Ubarnes, "Di-´ agnosis of horizontal pipe leaks using neural networks," *Journal of Physics: Conference Series*, vol. 1388, p. 012032, 11 2019.

[3] Vrachimis, Eliades, Taormina, Ostfeld, Kapelan, Liu, Kyriakou, Pavlou, Qiu, and Polycarpou (2020). "BattLeDIM: Battle of the Leakage Detection and Isolation Methods." BATTLEDIM 2020 https://battledim.ucy.ac.cy/