[Team 14]Water Leakage Detection using Neural Networks

Shreya Sabu
ssabu2@ncsu.edu

Krupal Shah
khshah2@ncsu.edu

Vedashree Chaphekar
vschaphe@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. We address the problem of demand for well equipped and maintained homes by proposing a cost effective solution to detect leakages and manage pressure, which in turn leads to significant water savings and reduced pipe breakage frequencies, especially in older infrastructure systems.

This is a research project under Dept of Civil, Construction, and Environmental Engineering. Developed two different models, the first one, which is a classification model, is based on Supervised Self-Organising Maps using SuSi framework, which can perform unsupervised, supervised and semi-supervised classification and regression tasks on high-dimensional data. The second model, which is a regression model, is based on Multi-Layer Perceptron (MLP) algorithm, which is a class of feedforward Artificial Neural Networks (ANNs).

The results of the evaluation can be summarized in three major findings:

- Both models were able to correctly classify the leak nodes with good accuracy.
- The Multi-Layer Perceptron model outperforms the Self-Organising Maps model.
- There isn't always need of a complex model. Simple MLP model achieve better accuracy as compared to complex SOM model.

I. Model Training and Selection

The objective of the project is to implement the leakage detection and analysis using Multi-Layer Perceptron Algorithm and Self Organizing Maps. 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. Developed two models, one a classification model based on Self Organizing Maps, and other a regression model based on Multi-Layer Perceptron Algorithm. Model based on Supervised Self Organising Maps is a prototype model, which was developed in order to observe whether it can learn the connection between leak nodes and sensor nodes. The main approach is the model based on Multi-Layer Perceptron Algorithm.

Data Generation: The data is in the form of leak values and pressure values which are simulated using the EPANET software used to model the water distribution systems [1]. 16 leak nodes and 16 pressure sensors are considered to generate dataset for the model. The data is in the form of leak values and pressure values in the leak dataset and pressure dataset respectively. The network consisting of leak nodes and pressure sensors is given in the Fig. 1. The leak dataset given in Fig.2 contains leak values and responses for leaks of up to 5 with random sizes ranging from 1 to 10. This is done for 10,000 realizations. Similarly, the pressure values represent the pressure drop at the sensor nodes due to the leaks present.

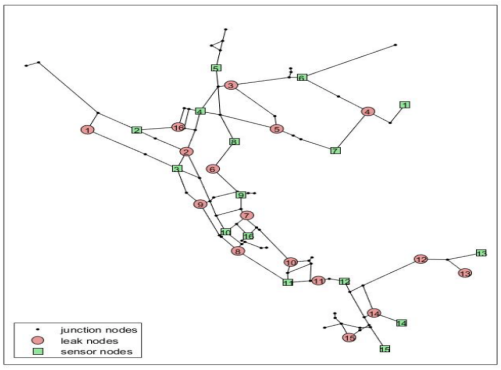

Fig. 1. Water Pipes Network

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 for 16 Nodes

The final dataset, which was generated, for leak values has a shape of 10000*16, and the dataset for pressure

values has a shape of 10000*16.

Self Organising Maps (SOMs): The Self-Organising Maps(SOMs) was introduced by Kohonen (1995). A self-organizing map (SOM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional, discretized representation of the input space of the training samples, called a map. It is a shallow ANN architecture consisting of an input layer and a 2-dimensional (2D) grid as output layer. The latter is fully connected to the input layer. Besides, the neurons on the output grid are interconnected to each other through a neighborhood relationship. Changes on the weights of one output neuron also affect the neurons in its neighborhood. This unique characteristic decreases overfitting of the training datasets. Self-organising maps differ from other artificial neural networks as they apply competitive learning as opposed to error-correction learning (such as backpropagation with gradient descent), and also they use a neighborhood function to preserve the topological properties of the input space.

Generally, Self Organising Maps are used for unsupervised learning tasks. Recent developments have been made for creating supervised and semi-supervised Self Organising Maps for the purpose of classification and regression. Developed a classification model based on supervised Self-Organising Maps, which will identify the respective leak node based on pressure values, using SuSi framework [2]. The SuSi framework is written in Python and it includes a fully functional Self Organising Maps for unsupervised, supervised and semi-supervised tasks. The main idea behind supervised SOM, which is taken from [2], is to attach a second SOM to the orginal unsupervised SOM. The two SOMs differ with respect to the dimension of the weights and their estimation algorithm. The weights of the unsupervised SOM have the same dimension as the input data. Thus, adapting these weights often changes the BMU for each input datapoint. In contrast, the weights of the supervised SOM have the same dimension as the target variable of the respective task. In the classification case, the weights contain a class. By combining the unsupervised and the supervised SOM, the former is used to select the BMU for each datapoint while the latter links the selected BMU to a specific estimation.

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, 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 [3] describes model based on Self Organising Maps(SOMs) and Multi Layer Perceptron Algorithm (MLP) for detection and localisation of water leaks. They experimented with different models based on SOMs and MLPs for the task of detection of leaks, they were able to achieve good accuracy in between 73% to 85%.

The detection of the presence of leaks in pipeline transport systems using Multilayer Perceptron is discussed in the paper [4]. 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 as follows:

Libraries	Functions
sklearn	Data preprocessing, accuracy measure, predicting probabilites, confusion matrix etc
SuSi	Implementation of Supervised Self Organising Maps
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.

A. *Model*

Implemented two models, classification mode based on Supervised Self Organising Maps, and a regression model based on Multi Layer Perceptron Algorithm. Both models follows a general network architecture pipeline which is given in Fig 3.

Self Organising Maps Model: The main objective is to classify a node as a leak node or not based on pressure values. Self Organising maps have been used for many times, for the problem of detection of water leaks in a smart water system, because of it's capability to produce a low-dimensional discretized representation of the input space. [3] [5]. The model has shown good results in detection of leaks in smart water system. Classification model based on supervised Self-Organising Maps using SuSi package is implemented. [2].

The raw pressure data has pressure values for each pressure node. The pressure values were normalized in order to have a common scale for all pressure values. The raw leak

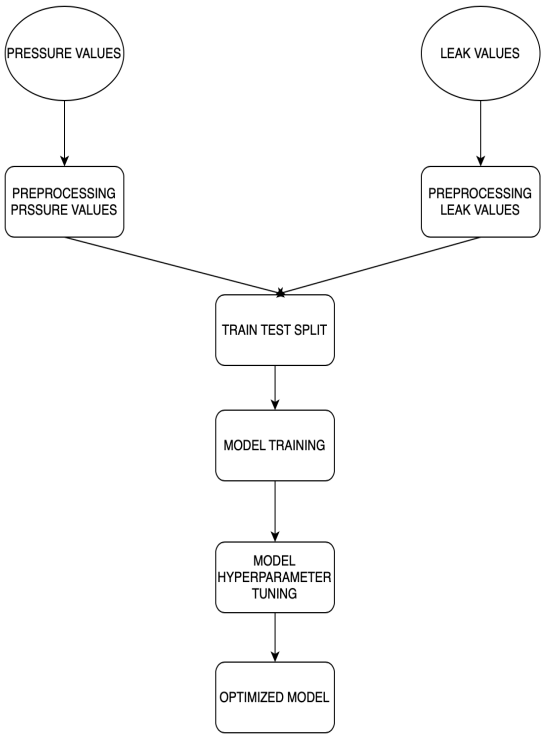

Fig. 3. Network Architecture

node data has leak values for each leak node in the network. The leak node is the node with the highest leak value, and for each set of pressure value, we extracted the respective leak node. Both the datasets are processed and created a dataframe set of pressure values and its corresponding leak. The dataset is split into training and testing dataset with the split ratio as (80,20). The SOM model is then trained using the training data, and classification is done on the test data using the trained model. The input to the model were the pressure values and the output was the respective leak node. The model is evaluated using the accuracy metric, i.e. the number of correctly classified instances and compared different models based on that. The hyper parameter tuning is done which is described in detail in "Model Selection" section. The final optimised SOM model has the following structure:

- n_rows: 150
- n_columns: 150
- n_iter_unsupervised(N.I.U): 12000
- n_iter_supervised(N.I.S): 9000
- learning_rate_start(L.R.S): 0.6
- learning_rate_end(L.R.E): 0.7

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.

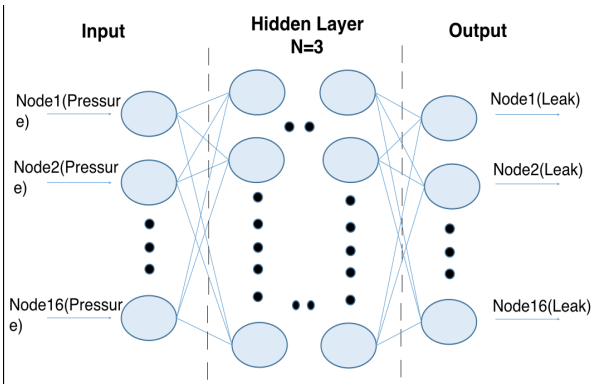

Fig. 4. Multilayer Perceptron Model

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. The table below 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

B. Baseline

Researching about this topic and discussing with Dr. Mahinthakumar, we came to the conclusion that a good comparison baseline model will be, MLP 10-14- 1 for the MLP based model with a baseline accuracy of around 85.4% , and SOFM 11-100 for the SOM based model with a baseline accuracy of around 64% . [6], [5], [3]

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 classification models were evaluated and compared, which are based on Self Organising Maps, based on accuracy. Also, 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*

Self Organising Maps (SOMs) Model: Implemented classification model based on SOMs using SuSi package [2]. Hyperparameter tuning was done on every hyperparameter and the final parameters were chosen based on highest accuracy achieved by a model. Hyperparameters, for which we performed tuning, related to supervised this supervised self organising maps are as follows:

- n_rows: Number of rows for the SOM grid.
- n_columns: Number of columns for the SOM grid.
- n_iter_unsupervised(N.I.U): Number of iterations for the unsupervised SOM.
- n_iter_supervised(N.I.S): Number of iterations for the supervised SOM.
- learning_rate_start(L.R.S): Learning rate start value.
- learning_rate_end(L.R.E): Learning rate end value.

The major results for the model are tabulated as follows:

rows	cols	N.I.S	N.I.U	L.R.S	L.R.E	Accuracy
50	50	1000	1000	0.5	0.5	34%
100	100	2000	2000	0.5	0.6	54.5%
75	75	4000	4000	0.7	0.9	62.5%
100	100	7000	7000	0.8	0.8	70%
50	50	6000	9000	0.8	0.8	70.5%
125	125	7000	7000	0.8	0.5	71%
125	125	9000	9000	0.5	0.7	77%
150	150	12000	9000	0.6	0.7	79%

The row in the above table represents the default configuration and the accuracy in that configuration. From the table above, it is observed that the accuracy of the model increases as the model complexity increases. The graph in Fig 5 shows that the accuracy increases with the increase of number of supervised iteration.

The graph clearly shows that number of iteration of the supervised SOM has a major impact on the performance of the model. The highest accuracy achieved after hyperparameter tuning is is 79%.

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

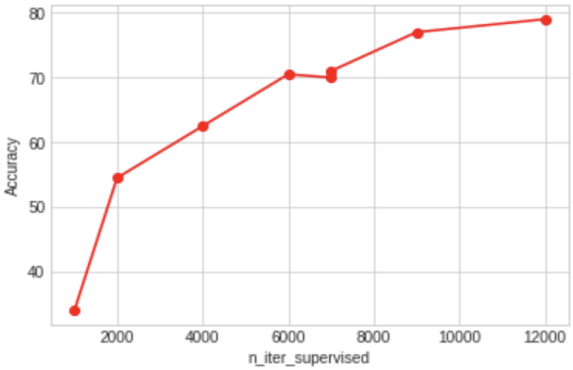

Fig. 5. Accuracy vs n_iter_supervised

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.

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

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. This threshold was later varied from 0.1 to 0.9 in steps of 0.1 and the accuracy was varied accordingly. 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.

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

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. 6 clearly shows that the mean absolute error is decreasing as the total number of epochs were increasing.

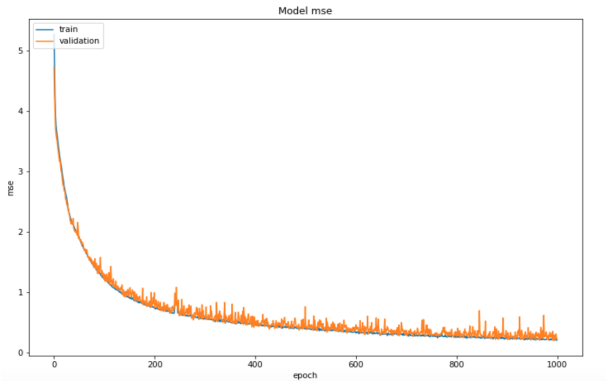

Fig. 6. Model Loss of Final Model

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

C. *Performance and Comparison to the Baseline*

Self Organising Maps: The highest accuracy achieved by the SOM model after hyperparameter tuning is 79%. It can observed the accuracy increases as the model becomes more complex.

Model	Accuracy
SOM 11-100 (Base)	64%
Supervised SOM (Proposed)	79%

Multi-Layer Perceptron: The highest accuracy obtained for the MLP after hyper parameter tuning was 97.23%.

Model	Accuracy
MLP 10-14-1 (Base)	75%
MLP Model (Proposed)	97.23%

It can be observed that the MLP based model clearly outperforms the model based on SOM. Also, observation can be made about the complexity and performance, it can be observed that the performance achieved by the complex SOM based model is similar to the performance achieved by a less complex MLP based model. It can also be concluded that as the number of classes increases the performance of the Supervised SOM decreases.

REFERENCES

[1] [Online]. Available: https://wntr.readthedocs.io/en/latest/

[2] F. M. Riese and S. Keller, "SUSI: supervised self-organizing maps for regression and classification in python," *CoRR*, vol. abs/1903.11114, 2019. [Online]. Available: http://arxiv.org/abs/1903.11114

[3] 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

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

[5] I. Barradas, L. E. Garza, R. Morales-Menendez, and A. Vargas-Martínez, "Leaks detection in a pipeline using artificial neural networks," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, E. Bayro-Corrochano and J.-O. Eklundh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 637–644.

[6] Y. Jin, W. Yumei, and L. Ping, "Approximate entropy-based leak detection using artificial neural network in water distribution pipelines," in *2010 11th International Conference on Control Automation Robotics Vision*, Dec 2010, pp. 1029–1034.