INCREASING THE ROBUSTNESS OF CLASSIFICATION ALGORITHMS TO QUANTIFY LEAKS THROUGH OPTIMIZATION

Ary Mazharuddin Shiddiqi
Department of Informatics, Institut Teknologi Sepuluh Nopember
e-mail: ary.shiddiqi@if.its.ac.id

ABSTRAK

Kebocoran dalam jaringan pipa air telah menelan biaya miliaran dolar setiap tahun. Untuk mengatasi masalah ini, diperlukan metode kuantifikasi kebocoran yang handal (untuk mendeteksi dan melokalisasi) untuk meminimalkan kerugian. Kami mengkuantifikasi kebocoran pada jaringan distribusi air dengan mengklasifikasikan lokasi kebocoran menggunakan algoritma pembelajaran mesin, yaitu Support Vector Machine dan C4.5. Algoritma tersebut dipilih karena kinerjanya yang tinggi dalam klasifikasi. Kami mensimulasikan kebocoran pada posisi dan ukuran yang berbeda dan menggunakan data simulasi tersebut untuk melatih algoritma pembelajaran mesin. Kami mengatur parameter dari algoritma-algoritma tersebut dengan mencari parameter yang paling optimal dalam proses pelatihan. Kemudian, kami menguji model algoritma terhadap data pengamatan nyata. Kami juga bereksperimen dengan data derau, karena ketidakakuratan sensor, yang sering terjadi dalam situasi nyata. Terakhir, kami membandingkan kedua algoritma tersebut untuk menyelidiki seberapa akurat dan handal dalam melokalisasi kebocoran dengan data yang derau. Kami menemukan bahwa C4.5 lebih handal terhadap data derau daripada SVM.

***Kata Kunci**: jaringan distribusi air, kuantifikasi kebocoran, penambangan data, sensor.*

ABSTRACT

Leaks in water pipeline networks have cost billions of dollars each year. Robust leak quantification (to detect and to localize) methods are needed to minimize the lost. We quantify leaks in water distribution networks by classifying their locations using machine learning algorithms, namely Support Vector Machine and C4.5. The algorithms are chosen due to their high performance in classification. We simulate leaks at different positions at different sizes and use the simulation data to train the machine learning algorithms. We tune the algorithms by optimizing the algorithms' parameters in the training process. Then, we tested the algorithms' models against real observation data. We also experimented with noisy data, due to sensor inaccuracies, that often happen in real situations. Lastly, we compared the two algorithms to investigate how accurate and robust they localize leaks with noisy data. We found that C4.5 is more robust against noisy data than SVM.

***Keywords**: data mining, leak quantification, sensor, water distribution networks.*

I. Introduction

OUT of twelve water utilities in Western Australia during the financial year from July 2012 to July 2013, six reported increases in the number of pipeline leaks, four reported decreases and only two remained steady [7]. The average amount of water leak of these twelve utilities reached 19% of the total water supply which costs billions of dollars each year. Therefore, it is crucial to detect leakage in a pipeline network.

Leak quantification consists of three sub-problems: detecting the presence, localizing the position, and predicting the size of leaks. Previous techniques were developed based on mathematical modelling [8], classification technique [4]–[6], and artificial neural network [3]. Mathematical modeling is difficult to implement on large scale networks due to modeling capacity, while ANN requires large computational resources to perform the task. Therefore, classifications algorithm have a better potential for developing techniques in this field.

Leaks can be located anywhere in a pipeline or on junctions. The presence of a leak causes the change of flows in pipes that can be used to identify leaks. A simple way of detecting leaks is by using the water balance principal. Each flow going into a pipeline network should be equal to the one(s) leaving the network. If not, a leak is suspected in the network. The water balance principal can only be applied If all flow sensors are correctly functioning. However, real-world sensors are not ideal. It has a certain degree of accuracy range. In this work, we observe how such inaccurate sensors can affect the performance of leak classification.

We model the sensor inaccuracy by creating a noise profile injected to the sensor data. Then, we investigate which of the two algorithms (SVM and C4.5) that performs better to handle noisy data. We address class imbalance problem in multi-class SVM that can lead to bias decision due to the tendency of classification to a larger tree side. This paper is structured as follows: First, a literature review is discussed to explore the previous research in leak quantification technique. Then, we present our research method and then followed by experiment results and analysis. Finally, we present the conclusion and future works

DOI: http://dx.doi.org/10.12962/j24068535.v18i1.a841

II. Literature Review

Classification-based leak quantification technique can be performed by labeling the pipeline network as two classes: leaky or not. Then, the locations of leaks are subsequently marked as sub-classes under the leaky class. Previously developed classification-based leak quantification techniques used flow, pressures, transient data for training and testing purposes [4]–[6], [11], [12]. Flow sensor is used in [4], [5], transient pressures are used in [11], while pressure data is used in [6], and both data are used in [12]. Our research uses flow sensors to quantify leaks due to its characteristic that is less sensitive to the fluctuation of flow.

Mamo et al. [5] used multi-class SVM which is an extended version of original SVM that can only handle two classes. The multi-class SVM labels a suspected node with +1, while the other unsuspected nodes as 1. The research produced a promising performance in classifying leaks in a District Meter Area (DMA) based water distribution network. To maximize the performance of the technique, the research suggests using data gathered from a real environment with advanced multi-parameter monitoring sensors. However, Mamo et al. did not address the problem when noise are present in data.

The use of accurate data, provided by sensors, determines the performance of leak quantification techniques. Data can be susceptible to noise due to interference from surroundings or sensor inaccuracies. The latter cause often happens in real situations as aging sensors often could not maintain their functionality. Commonly, the noise level of sensors can be retrieved from the datasheet from the sensors' manufacturer. For example, a sensor has a range of inaccuracy at ± 3% [1], which means that the sensor's performance can drop up to 3% inaccuracy at the worse level.

Saez et al. [10] have investigated the effect of noise on three popular classification algorithms, i.e. C4.5, k-Nearest Neighbors and SVM. The research used Uniform and Gaussian distribution algorithm to perturb training datasets in their experiments. The research found that SVM performs best than k-NN and C4.5 with noise-free data, while C4.5 performed best in handling the effect of noise on data compared to k-NN and SVM. Also, the research found that the uniform distribution has a worse effect on dataset compared to the gaussian distribution due to the scattered pattern of the uniform distribution.

III. Methodology

Our leak quantification technique starts by designing an operational model of a pipeline network (Figure 2) simulated using EPANET [9]. EPANET contains parameters of the network such as pipes, junctions, pressures, etc. that influences how the hydraulic system works. We use a pipeline network used Pudar & Ligget [8] and Cardell-Oliver et. al. [4] (Figure 1) as the baseline of our research. The network consists of a reservoir (100r), six real nodes (1,2,3,4,5,101,102), four virtual nodes to simulate leaks along pipes (6,7,8,9), and demand nodes (101d,102d). Baseline data are generated from the network's initial operational configuration to obtain leak free condition of the network.

To characterize leak signatures, we recorded flow changes triggered by leaks at all possible positions in the network with various sizes. We used a Python script to introduce leaks by modifying emitter coefficient at junctions of the operational network. Then, EPANET tool is called through Python *subprocess* to generate a new scenario. Each time the *subprocess* is called, the EPANET tool will simulate an operating pipeline model based on the scenario generated. Flow and pressure changes due to leaks at different positions vary due to the hydraulic system of the network (Table I). This process is performed until leaks are simulated at all possible positions. The magnitude of flow change at the inlet of the network (P100) equals the size of the leak, while the flow changes in pipes can be used to characterize leaks. We record all flow data from the scenarios in a *csv* file.
To simulate noisy data, we use R statistical software and apply a Gaussian function to perturb flow data. This procedure gives us a range of values within the "bell-shape" of the Gaussian function. The variation of the perturbed flow could easily cause the performance of a classification model to fall. A classification algorithm has some parameters that drive the creation of a model. When these parameters are changed, then the classification model generated will differ. We argue that the effect of noise in data can be minimized by finding optimum parameters of the classification algorithms. Hence, the performance of the algorithms can be maintained.

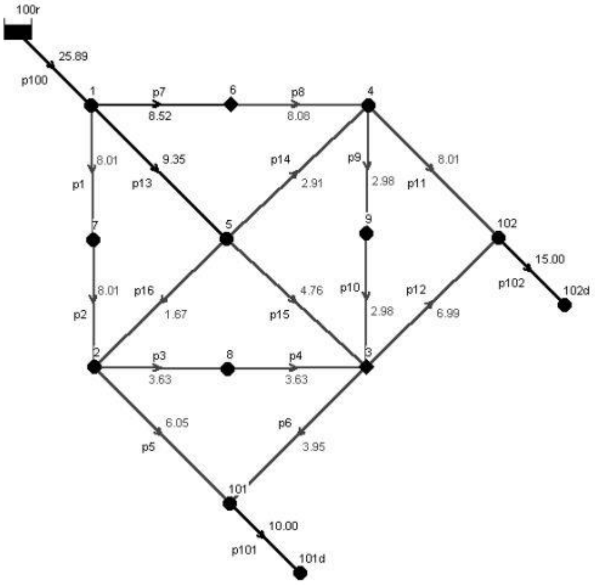


Fig. 1. An example of pipeline network structure with one tank, two demand points, seven actual junctions and four artificial junctions[4], [8].

IV. Locating Leak Position Using Classification Technique

We designed specific methods to classify leak location based on the characteristics of SVM and C4.5. With SVM, first, we categorize a water distribution network in a leaky condition or not. Then, should the network is classified as leaky, a nested classification mechanism is placed under the leaky class to locate the actual leak location. This method is performed until a junction, suspected as the source of the leak, is found. While with C4.5, we generate a classification tree where the branches represent the classes (locations) of leaks.

A. Support Vector Machine (SVM)

SVM creates a classification model based on support vectors. The support vectors are the training-data points of any class that can be used as features to construct a hyperplane with maximal distances to separate classes the further the distance, the lower the probability of miss-classification.

In linearly separated classes, the hyperplane construction is straightforward. While in non-linearly separated classes, the functionality of support vector machines needs to be extended to nonlinear classifiers. This can be done by applying a kernel trick to a linear support vector machine. Selecting an appropriate kernel and the associated parameter values, such as the degree of cost and gamma for the RBF kernel, is called a feature search. This task is not trivial, and there will be some trade-offs in model complexity and performance.

As SVM relies on the support vectors in classification, noise in a dataset could cause a complicated boundary between classes. Therefore, the hyperplane constructed will not be accurate which can lead to a wrong classification model. Further investigation of the effect of noise on SVM performance will be presented in Section VI.

B. C 4.5

C4.5 is a classification algorithm that uses a decision tree from a given training set of non-categorical attributes (A1, A2, ..., An) to correctly predict the value of the categorical attributes (C1, C2, ..., Cn). In the decision tree, each node corresponds to a non-categorical attribute and each edge corresponds to a possible value of that attribute. A leaf of the tree specifies the expected value of the categorical attribute for the records which is described by the path from the root to that leaf. Each node should be associated with the non-categorical attribute which is the most informative among the attributes not yet considered in the path from the root. Entropy is used to measure how informative a node is.

The decision tree generated from a training set may become quite complex with long and very uneven paths. A pruning strategy is applied to reduce the complexity of a decision tree model and to improve classification accuracy. The pruning technique reduces the size of a decision tree model by removing sections with less contribution to the model. The pruning strategy is performed by representing a whole sub-tree with a leaf node. This process is done when an expected error rate in the sub-tree is greater than in the single leaf.

TABLE I
FLOWS AND PRESSURES PRODUCED FROM SIMULATE LEAKS AT DIFFERENT SIZES [0.1, 1.0] LPS AND AT DIFFERENT SIZES.

Pipe	Flow value range (lps)	Node	Pressure value range (psi)
p1	[7.6, 9.36]	node 1	[19.78, 19.83]
p2	[6.97, 8.77]	node 2	[19.63, 19.72]
p3	[3.12, 4.65]	node 3	[19.6, 19.69]
p4	[2.42, 4.09]	node 4	[19.62, 19.71]
p5	[5.73, 6.39]	node 5	[19.64, 19.72]
p6	[3.61, 4.3]	node 6	[19.69, 19.77]
p7	[7.88, 9.64]	node 7	[19.7, 19.77]
p8	[7.24, 9.05]	node 8	[19.61, 19.7]
p9	[2.41, 4.04]	node 9	[19.61 , 19.7]
p10	[1.76, 3.52]	node 101	[19.58 , 19.67]
p11	[7.68, 8.37]	node 101d	[19.55 , 19.64]
p12	[6.66, 7.33]	node 102	[19.53 , 19.62]
p13	[8.85, 10.23]	node 102d	[19.46 , 19.56]
p14	[2.23, 3.66]	node 100r	
p15	[4.35, 5.33]		
p16	[1.01, 2.6]		
p100	[9.75, 10.25]		
p101	[14.63, 15.37]		
p102	[24.39 , 27.7]		

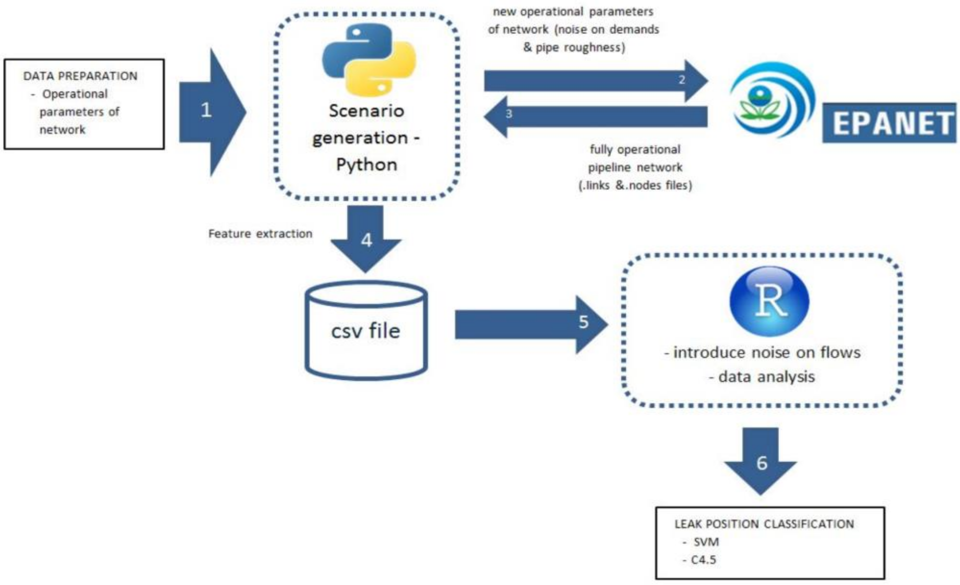

DATA PREPARATION
- Operational parameters of network
1
Scenario generation - Python
new operational parameters of network (noise on demands & pipe roughness)
2
3
EPANET
fully operational pipeline network (.links &.nodes files)
Feature extraction
4
csv file
5
R
- introduce noise on flows
- data analysis
6
LEAK POSITION CLASSIFICATION
- SVM
- C4.5

Fig. 2. Leak quantification process.

V. NOISE MODELLING

Imagine that data is in tabular format (let us call it the *leak influence matrix*), the columns represent sensor readings or attributes and the rows represent data points produced by the sensors. Each attribute corresponds to each sensor. Perturbing the values in columns of the table by using the Gaussian function can be regarded as noisy data due to sensor inaccuracies. The Gaussian function is modeled as stated in Equation 1.

$$\sigma = \frac{V\ x\ \eta}{\kappa} \tag{1}$$

where σ is the perturbed value of flow in a cell of the leak influence matrix, V is the actual value of each data point, η is the level of noise used, and κ is Gaussian standard deviation. κ determines the scope of Gaussian distribution. For example, if the value is set to be 3, then the coverage is ± 99.7% of the Gaussian bell-shape. The lower the κ value, the fever coverage of the distribution.

To observe the effect of noise in the classification process, we use a different level of noise from 0% – 5% with 1% increments. This value is used to describe sensor inaccuracy where 0% symbolizes that there is no noise in the data, and 5% symbolizes the worst scenario where a sensor is very inaccurate. This value is more than the average sensor inaccuracy as stated in [2], that is ± 3%. The noisy data is tested against clean data and correlated noisy data to observe how much classification performance is affected.

VI. Experiment Results

We used the E1071 and RWeka libraries in R to implement SVM and C4.5 classification algorithms respectively. We measure the performance of the classifiers by using precision, recall, and F measure metrics.

A. Experiment Scenarios

Based on the characteristic of SVM that can only handle binary classification, to implement a multi-class classification, we used one vs all or hierarchical classification strategy. On the other hand, C4.5 can handle multi-class classification directly. Therefore, leak localization is performed by finding the furthest down match leaf(s) in the tree.

We use the same data for both SVM and C4.5 classification experiments, i.e., noise-free and noisy data. As explained in Section V, the noise is gradually introduced to clean data from 0% up to 5%. This approach is aimed to observe how much the noise affect the classification performance at every level. We used two schemes for testing our classification models, i.e., clean data vs noisy data and noisy data vs noisy data. The first scheme is aimed to simulate the real-world situation where training data is taken from noise-free sensors, while the testing data is taken from sensors with possible inaccurate readings. The second scheme is to depict real situations where sensors can be inaccurate, and testing data is also taken from inaccurate sensors. The second scheme is aimed to examine the robustness of our approach when it is not possible to replace inaccurate sensors with the correct functioning ones due to a limited budget.

1) SVM one vs All classification strategy.

The one vs all strategy is applied using leak or no leak labels stated as the value of ’1’ or ’0’ where one means the state falls to the side and 0 is vice versa. Then, we labeled each sub-tree as 1 should a leak is detected on the sub-tree side. We generated up to 100 leak sizes [0.1,1.0] liter per second at every location to validate our approach. Thus the total of 1800 leak cases for nine possible locations. A 10-fold cross validation strategy was used for training and testing stages.

2) Hierarchical classification strategy

The hierarchical strategy is used to divide the leak locations into a 2-leaf tree for each layer. This strategy is used to minimize the steps needed to locate a leak. The difference with the one vs all strategy is in the way to divide the class clusters. For each level of the hierarchical classification, each cluster from the upper level is divided close to equal partitions (Figure 3). This process continues until each class becomes a leaf of its upper level. The benefit of this strategy is to have a more balanced training and testing data.

3) C 4.5

C4.5 is a natural multi-class classification algorithm. A tree traversal method is performed to locate leaks by matching the values of an observed data to the edges of a node. The tree traversal stops when it ends with a single label on a leaf. The same data is used as in the classification using SVM classifier, and 10-fold cross-validation was also applied.

B. Parameter Sweeping

Parameter sweeping technique is used to find the best parameters combination for classifiers. Tuning SVM is by modifying cost and gamma parameters (Figure 4(a)), while the performance of C4.5 is adjusted by setting minimum object and confidence level (Figure 4(b)). Referring to this experiment, the best parameters for SVM are cost = 1 and gamma = 1 and C4.5 are Min Object = 4 and Confidence factor = 0.1. The optimized parameters of the algorithms are then used for leak quantification experiments.

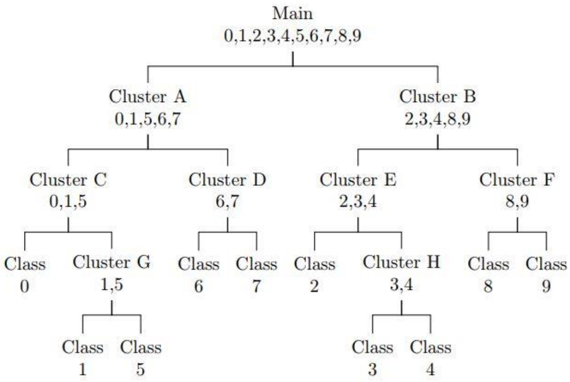
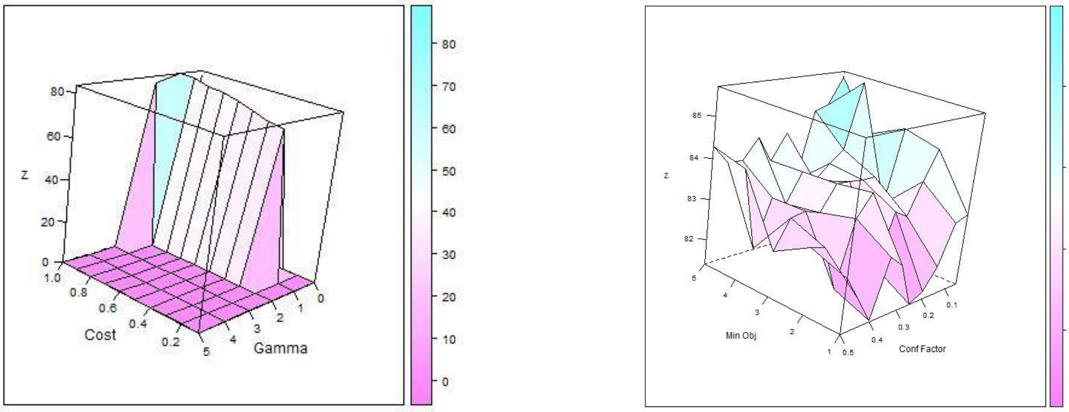


Fig. 3. Hierarchy tree of SVM.



(a) SVM parameter sweeping

(b) C 4.5 parameter sweeping

Fig. 4. Parameter sweeping results: (a) SVM Parameter Sweeping Cost = [0.1 , 1] and Gamma = [0 , 5] with best choice is Cost = 1, Gamma = 1 (b) C4.5 parameter sweeping = MinObj = [1 , 5] and Confidence Factor = [0 , 0.5] with best choice is MinObj = 4, Confidence Factor = 0.1.

C. Experimental Results

We generated classification models using the algorithms and tested the models to examine their performances. Each experiment used noise-free and noisy data for testing purpose.

1) Randomized parameter vs optimized parameter

As the baseline of our experiments, we compare the performance of SVM and C4.5 algorithms to quantify leaks with randomized and optimized parameters. Our experimental results show that the algorithms with optimized parameters outperform the corresponding algorithms with randomized parameters (Figure 5(a) and 5(b)). The results indicate the importance of finding the proper parameters for a set of data.

2) SVM one vs All

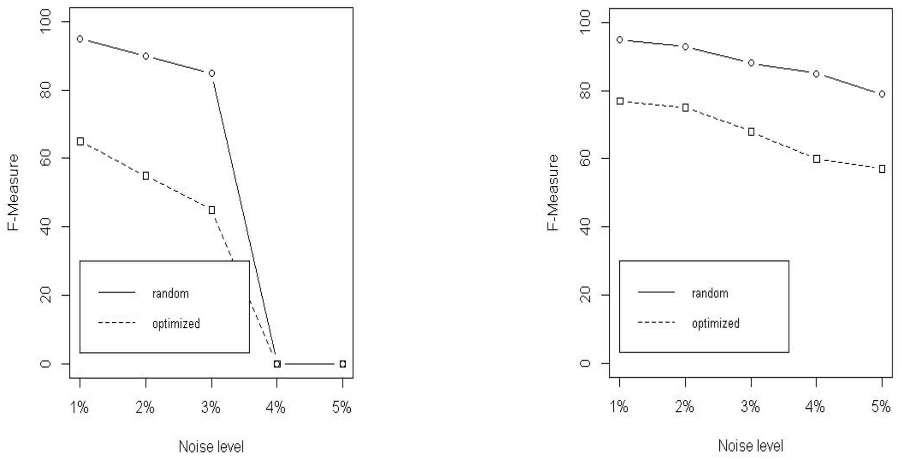
As expected, the class imbalance in this scenario has caused the classification performance drop significantly (Figure 6(a), 6(b) and 6(c)). After the noise level is above 2%, each performance evaluation of this strategy has fallen to less than 10%. This result shows that the one vs all strategy is not robust against noisy data.

3) SVM Hierarchy

This strategy is aimed to overcome the weakness of the one vs All strategy due to class imbalance. The experiment results (Figure 7(a), 7(b), and 7(c)) have shown that this strategy was successful at a certain degree. As the noise increases, the performance started to decrease. However, the decrease was more gradual rather than a dramatic slope as in the one vs all strategy. A significant improvement is at the recall performance on clean training data vs noisy testing data that can be maintained by over 20% at the highest noise level (5%).

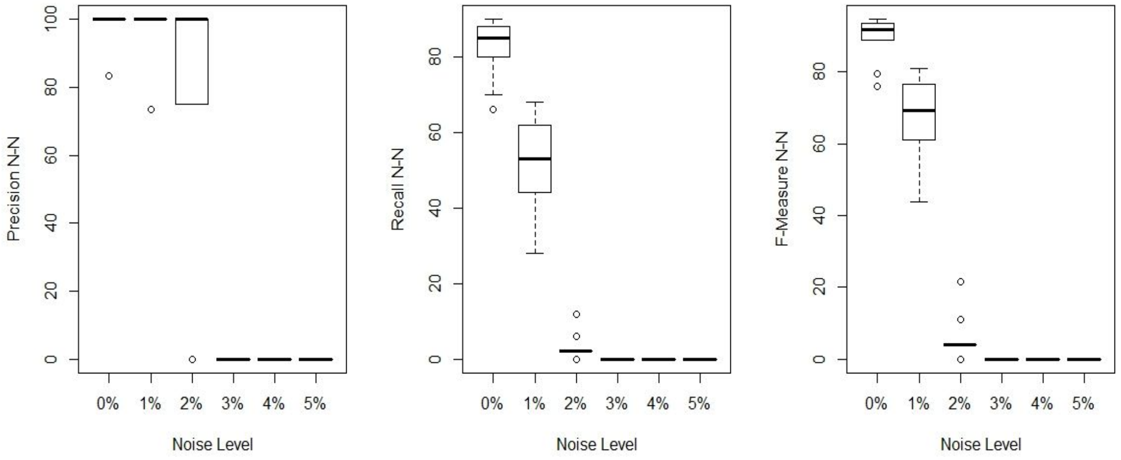
4) C4.5

C4.5 performed well against noise-free and noisy data indicated by high precision, recall and F-measure scores Figure 8(a), 8(b), and 8(c)). The three measurements showed that the average scores of the C4.5 algorithm are ± 80% up to 4% noise level. The C4.5 overall scores are much higher compared to the SVM overall scores in both one vs all and hierarchy strategies. This result shows that C4.5 is a much better algorithm over SVM against noisy data. This result is important considering that noise can often present in real-world applications; even sensor manufacturers have been researching to improve the quality of sensors, but still, there are no such perfect sensors.



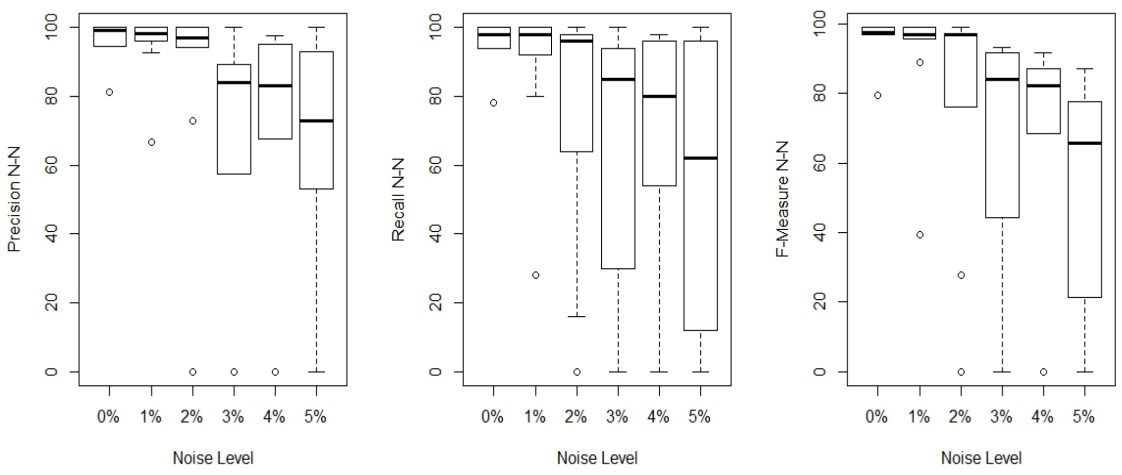
(a) (b)

Fig. 5. Performance comparisons of (a) SVM without optimized vs optimized parameters (b) C45 without optimized vs optimized parameters.



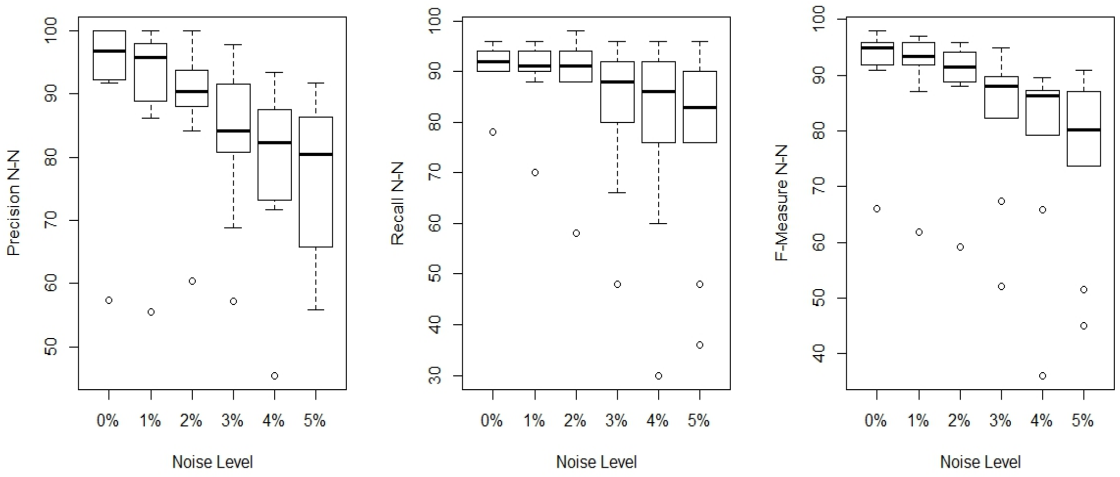
(b) Precision (b) Recall (c) F-Measure

Fig. 6. Performance measurements of SVM classifier when tested against noise free and noisy data.



(a) Precision (b) Recall (c) F-Measure

Fig. 7. Performance measurements of SVM classifier when tested against noise free and noisy data.



(a) Precision (b) Recall (c) F-Measure

Fig. 8. Performance measurements of C4.5 classifier when tested against noise free and noisy data.

VII. Conclusion

We presented a technique to find the optimized parameters of SVM and C4.5 algorithms to quantify leaks. We also investigated the effect of noise on the performance of the algorithms. Although SVM is a well-known classifier with high performance, we proved that it is susceptible to noisy data. The proposed hierarchical classification technique was able to lift the performance of SVM at certain levels. However, it is still not enough to prevent the declined performance of SVM against noisy data. On the other hand, C4.5 has proven that it is more robust against noisy data compared to SVM both the one vs all and hierarchical strategies. Therefore, we conclude that SVM would not be the best option to quantify leaks when noise is present.

We showed that the optimized parameters of an algorithm can improve its performance to quantify leaks for a set of data both in noise free and noisy data. However, for different set of data, the optimized parameters could be different. Therefore, we plan to find an empiric method to find optimized algorithm parameters for multiple datasets. Also, we would like to investigate further on the use of network structure information to improve the performance of leak quantification algorithms.

References

[1] 7000 / 8000 Series Flow Meter With Optional Outputs. Technical report, Clark Solutions.

[2] Typical surface roughness. Technical report, Engineering page, 2015.

[3] Ignacio Barradas, Luis E Garza, Ruben Morales-Menendez, and Adriana Vargas-Mart´ınez. Leaks detection in a pipeline using artificial neural networks. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5856 LNCS:637–644, 2009.

[4] Rachel Cardell-Oliver, Verity Scott, Thomas Chapman, John Morgan, and Angus Simpson. Designing sensor networks for leak detection in water pipeline systems. In 2015 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (IEEE ISSNIP 2015), Singapore, April 2015.

[5] Thewodros G Mamo. Virtual DMA Municipal Water Supply Pipeline Leak Detection and Classification Using Advance Pattern Recognizer Multi-Class SVM. Journal of Pattern Recognition Research, 1:25–42, 2014.

[6] J Mashford, D De Silva, D Marney, and S Burn. An approach to leak detection in pipe networks using analysis of monitored pressure values by support vector machine. In Network and System Security, 2009. NSS '09. Third International Conference on, pages 534–539, Oct 2009.

[7] Water Services Association of Australia. National performance report 2012–13: Urban water utilities. Technical report, Technical report, National Water Commission and Water Services Association of Australia, 2013.

[8] Ranko S Pudar and James A Liggett. Leaks in pipe networks. Journal of Hydraulic Engineering, 118(7):1031–1046, 1992.

[9] Lewis A Rossman et al. EPANET 2: users manual. U.S. Environmental Protection Agency, Cincinnati, 2000.

[10] Jose A Saez, Mikel Galar, Julian Luengo, and Francisco Herrera. Tackling the problem of classification with noisy data using multiple classifier systems: Analysis of the performance and robustness. Information Sciences, 247:1–20, 2013.

[11] Seshan Srirangarajan, Muddaser Iqbal, Hock Beng Lim, Michael Allen, Ami Preis, and Andrew J. Whittle. Water Main Burst Event Detection and Localization. Water Distribution Systems Analysis 2010, pages 1324–1335, 2011.

[12] Andrew J Whittle, Michael Allen, Ami Preis, and Mudasser Iqbal. Sensor networks for monitoring and control of water distribution systems. International