QATAR UNIVERSITY

COLLEGE OF ENGINEERING

TOWARDS ACCURATE LOSS PREDICTION IN KAHRAMAA WATER STATIONS

BY

EMAN JASSIM AL-JABER

A Thesis Submitted to

the Faculty of the College of

Engineering

In Partial Fulfillment

of the Requirements

for the Degree of

Masters of Science in Computing

June 2018

COMMITTEE PAGE

The members of the Committee approve the Thesis of Eman Jassim Al-Jaber defended on 12/03/2018.

_____
Khaled Shaban
Thesis/Dissertation Supervisor

_____
Dr. Nader Moslem Meskin
Committee Member

_____
Dr.Jihad Jaam
Committee Member

_____
Dr. Samir Elhedhli
Committee Member

_____
Prof. Ali Jaoua
Committee Member

Approved:

_____
Khalifa Al-Khalifa, Dean, College of Engineering

ABSTRACT

AL-JABER, EMAN, J., Masters:

June: [2018:], Masters of Science in Computing

Title: Towards Accurate Loss Prediction in KAHRAMAA Water Stations

Supervisor of Thesis: Khaled, B., Shaban.

Qatar General Electricity and Water Corporation (KAHRAMAA, KM) deployed and started operation of water Supervisory Control and Data Acquisition (SCADA) system in 2006. The aims of this SCADA system are to increase the control and pumping of water to customers and reduce the water loss of the network. SCADA collects sensory data such as reservoirs' inlet and outlet flows, reservoirs' levels, reservoirs' inlet water stock as well as status of valves and used pumps. These time-stamped data are periodically transmitted to several back-end servers for logging, storing, and processing. KM water network is composed of 35 connected stations; each includes from 3 to 12 reservoirs. Currently, KM lacks the ability to accurately forecast any water loss in the network, except by assuming that historical loses apply the same in future; causing inaccurate predictions.

Throughout the years, there has been an increasing interest in water loss prediction. Different techniques are used to analyze and forecast the water loss. These techniques are classified into three categories, which are: *statistical*, *machine learning* and *hybrid* modeling approaches. Statistical approach depends on fitting mathematical models to the observed data. However, these have a disadvantage of high noise error that prevents water leaks to be accurately detected and forecasted. In machine learning, water loss is predicted

through training of various models such as Support Vector Machines, Artificial Neural Networks and Random Forest. The hybrid approach combines two or more techniques from the previously mentioned approaches.

This thesis studies methods to accurately predict water loss in KM water stations. We adopt a knowledge discovery and data mining process and activities that include *data collection*, *data preprocessing*, *feature engineering*, *model training*, and *validation*. This is the first automated attempt for KM to predict future volumes of water to be lost. Moreover, several contributions are made to advance prediction accuracy including those related to data preprocessing (data aggregation, cleaning, and transformation), feature engineering (feature generation, data windowing), and model training where several models are optimized for high accuracy using statistically reliable evaluation (cross-validation). Experimental results show that the highest water loss prediction accuracy of the next hour, 12^{th} hour, and 24^{th} hour are 84.78%, 73.01%, and 71.66%, respectively. These results come with different settings and parameters tuning that are optimized for each case. Moreover, all of the above results surpass baseline models by 14.78%, 45.32%, and 11.50%, respectively, in accuracy.

DEDICATION

For my biggest supporters, my parents.

Thank you for your support, inspiration, and tolerance throughout my study.

ACKNOWLEDGMENTS

I would like to acknowledge and thank the following important people who have assisted and supported me throughout my master degree.

Firstly, I would like to express my gratitude to my supervisor Dr. Khaled Shaban, for his unweaving support, insight thought and guidance this master thesis.

I would like to thank all my family and close friends. You have all encouraged and believed in me. You have all helped me to focus on what has been a hugely rewarding and enriching process.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

In this chapter, we introduce the water loss prediction problem and discuss motivations for tackling it. We also introduce relevant background topics such as time series forecasting techniques, and machine learning. Finally, we state the main objectives and challenges of the thesis.

Motivations

Water covers around 70% of the earth's surface, and it is considered one of the most valuable provisions of human life. Drinking water is important for human's existence even though it does not offer organic nutrients. Water is used on a daily basis in almost all aspects of life. With one of the lowest levels of rainfall, Qatar depends on water from limited resources. Furthermore, with the country's rapid population and economic growths, water scarcity has become a concern of increasing importance (Qatar's Sustainable Development, 2012). According to Qatar National Development Strategy 2011-2016, Qatar's water distribution system network loses yearly around 30%–35% of its resource. This is a high loss rate compared to the standard acceptable yearly leakage percentage which is around 18% (General Secretariat Development Planning, 2011). The high percentage of water loss in Qatar is critical and it manifests negatively on the costs, resources and environment. Therefore, it is timely important to be able to estimate and predict water loss to take quick and necessary actions to reduce it.

Qatar General Electricity and Water Corporation (KAHRAMAA, KM)

KM was initiated in July 2000 further to the Emiri law to maintain and regulate the production of electricity and water to customers in Qatar. Nowadays the corporation has the right of being the sole distribution and transmission system operator and owner for the water and electricity sector in Qatar. KM main objective is to afford high quality and sustainable water and electricity for better living in Qatar. Moreover, by 2030, KM mission is to set a global benchmark for technological innovation, performance, social responsibility and environmental sustainability in water and electricity sectors.

KM water distribution network consist of various elements such as transmission pipes, water reservoirs, pumping station, water towers. These elements are crucial infrastructure to provide water to the consumers. KM purchases water and electricity from water production plants which are known as Independent Power and Water Providers that are mainly owned by Qatar Petroleum. The purchased water is transmitted through transmission lines to water reservoirs that consist of some flow meters, water tanks, and pressure transmitters. Then, according to customers' demand, the water pumped from the water pump stations to water tanks or directly to customers' tanks. Customers can be houses, farms, industries, hospitals, schools, etc. KM water distribution network is shown in *figure 1* below.

The overall KM water distribution network is controlled, monitored and analyzed by the National Water Control Center (NWCC) which is introduced in the next section.

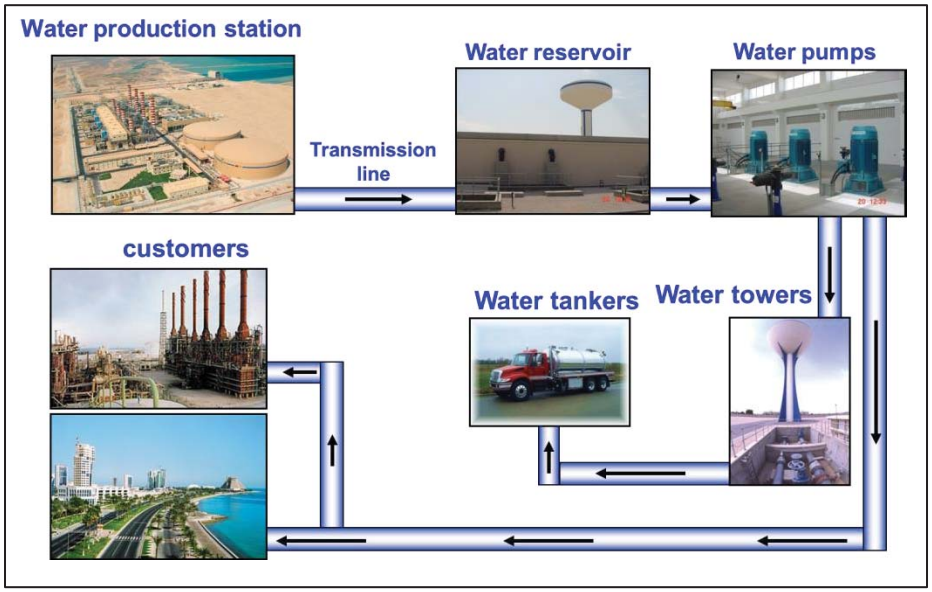

Figure 1. *KM water distribution network.*

National Water Control Center (NWCC)

The NWCC in Qatar is one of the powerful water monitoring and controlling systems in the Middle East. It started in 2009 using a SCADA system to ease the management, monitoring and controlling the whole water distribution network (WDN) and production in desalination plants. Nowadays, there are 35 water reservoirs and pumping stations and nine desalination plants all over the state of Qatar as shown in *figure 2*. KM has a daily total water stock of approximately 901.3 million gallons (MIG) and daily water distribution around 361.55 MIG. In general, the main responsibilities of SCADA engineers are to control, analyze and monitor the water forwarded from the desalination plant, water demand, reservoir capacity and water distribution. Moreover, they are also responsible for

monitoring the water flow, pressure, and water quality to give the customers the highest quality of water. Furthermore, Qatar is divided into different areas which are called District Metering Area (DMA) to facilitate the detection of any problem in a specific location such as water leak, quality, cutting, etc. In that regard, NWCC has to optimize the flow and pressure due to the demand change considering different factors such as holidays, seasons or weather.

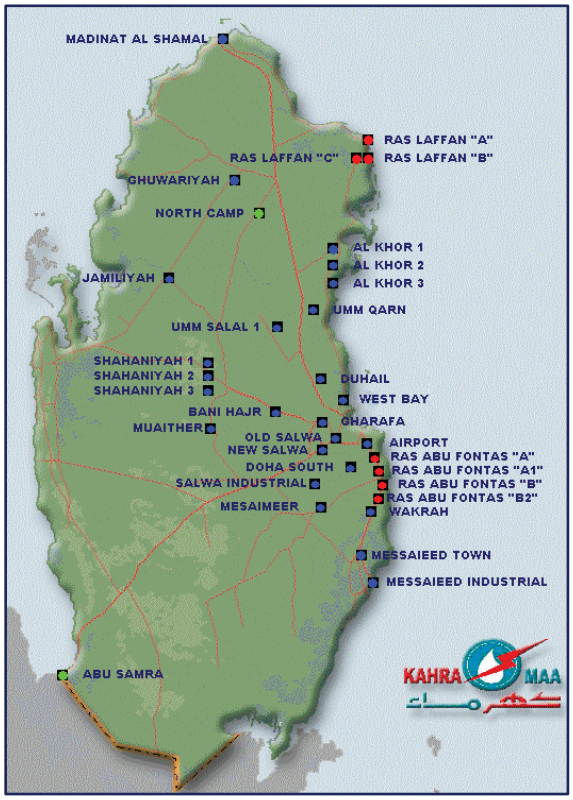

Figure 2. *Water reservoirs and desalination plants in Qatar.*

Figure 3 shows a snapshot from the SCADA system for one of KM water reservoirs and pumping stations, named "Airport Station". As noted in below figure, the station consists of several components including valves (1), flow meters (2), pumps (3), pipes (4), sensors (5), reservoirs (6), and pressure transmitters (7). Currently, controlling and monitoring the WDN. In that station as illustrated below has the following:

1. Two water inlets for water received by the desalination plants. Each inlet has pressure transmitter that shows the water pressure in that line, flow meter to represent the water flow in that line and water quality device to show the water qualities within that line.
2. Ten water tanks that show the water levels.
3. Nine pumps to pump the water received from the water tanks to DMA's.
4. Various number of valves that can be opened and closed by SCADA control engineer according to water demand and water forwarded to that stations.
5. Three outlet lines which have the same devices installed for the inlet lines. However, each outline line is forwarding water to one DMA site, so three outlet lines means three DMA customer sites.

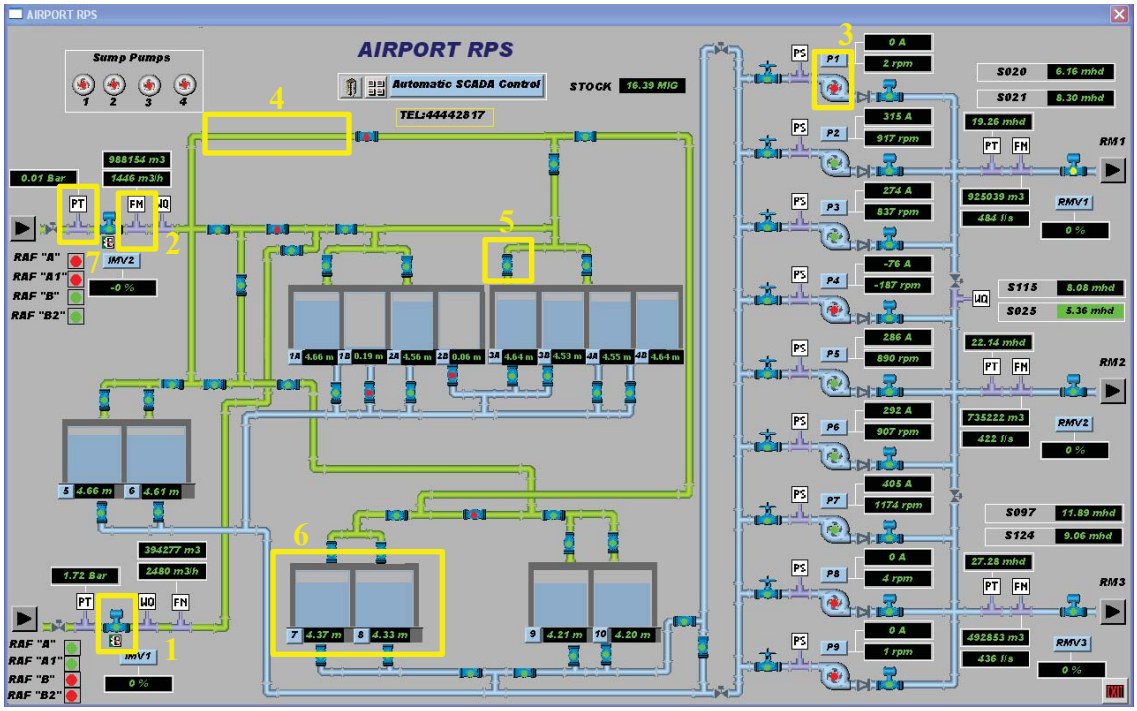

Figure 3. *Components of KM airport reservoir and pumping station.*

Time Series Forecasting

The readings received by KM SCADA system is a time-stamped data that are periodically transmitted to several back-end servers for logging, storing, and processing. This wealth of collected data, however, is not being utilized to its full potentials to foresee losses in order to prevent them.

Time series models have been the basis for any analysis and application of studying the behavior of process over a period of time. Times series models is a time oriented sequence of data (i.e., hourly, daily, and monthly) on a point of interest variable

(Montgomery, Jennings & Kulahci, 2015). The applications of time series models are diversified, such as weather prediction, sales prediction, inventory studies, etc. In decision-making associated with the uncertainty of the future, time series models is one of the most robust methods for forecasting. Mostly, future prediction and decision making for such processes will be based on what would be an expected result. The need for these expectations and predication has encouraged organizations to build forecasting techniques to be ready to face the apparently risky future. Also, time series models usually combined with data mining algorithm techniques to assist in understanding the behavior of the data and enable to forecast future patterns and trends in the data behavior (Montgomery, Jennings & Kulahci, 2015) (Box, Jenkins, Reinsel & Ljung, 2015).

Selecting the best forecasting technique is not the only aspect that affects prediction accuracy. Forecasting horizon is another aspect to consider that effects accuracy, where the horizon is the number of steps forecasted in the future. The higher the horizon, the more difficult the forecasting process is. Furthermore, another factor to look at when constructing the forecasting system is the number of variables, i.e., multivariate or univariate time series. In univariate time series problem only one variable is used, such as using the historical data of water loss to predict its future value. On the other hand, multivariate time series captures several variables, such as using the water input and output flow, stock level, and holiday seasons to predict the data of specific variable. This thesis investigates the efficiency of different machine learning techniques in forecasting time series of water loss. Different models and horizons and forecasting models for multivariate

time series data are investigated.

Machine Learning

KM lacks the ability to accurately forecast any water loss in the network by using any model, except by assuming that the exact historical loses apply the same in future; causing inaccurate predictions.

Kevin P. Murphy (2012) explained machine learning as an area derived from the broad area of Artificial Intelligence, which intends to simulate intellectual abilities of humans by machines. Specifically, machine learning takes into consideration the important question of how to make machines capable to ‘learn.' Learning in this framework is considered as an inductive interference, where one notices examples that show incomplete information about some data represented. Two learning approaches are known in this field: *supervised* and *unsupervised* learning. In supervised learning, there is a label represented in every example. It is assumed to be the answer to a question about the example. In case the label is discrete, then the learning process is aimed at solving a classification problem. For continuous labels, the problem to be solved is known as a regression problem. In unsupervised learning, one tries to reveal hidden regularities such as clusters or to find anomalies in the data like detecting unusual act in machine function, and this learning approach usually used in network intrusion (Murphy, 2012). In this study, we are investigating the supervised learning approach to solve the prediction problem. We are looking to answer the question “How much water loss is likely to accrue from a water station in next hour or next day?.” We are building models that learn from the time series

data collected from SCADA in one of the KM water stations.

The process in the above steps to apply and construct ML-based models for forecasting values of unseen target data is illustrated in *Figure 4*. In the training part, data with known target values are collected; a subset of features is selected, and then used to develop a prediction model. There are several subsets of features selected and different ML algorithms used; therefore, there are different predictors that can be trained. In the testing part, the produced models from the training part are validated and evaluated. Various methods are used in model validation.

In the deployment part, the best model and features will be used to process unseen data and generate prediction results. The model performance is kept on check to validate its prediction results. Basically, in changing environments, the process of training, testing, and deployment are regularly repeated to achieve high accuracy of results. This iterative process can be carried-out to improve performance of the models as historical data become increasingly available.

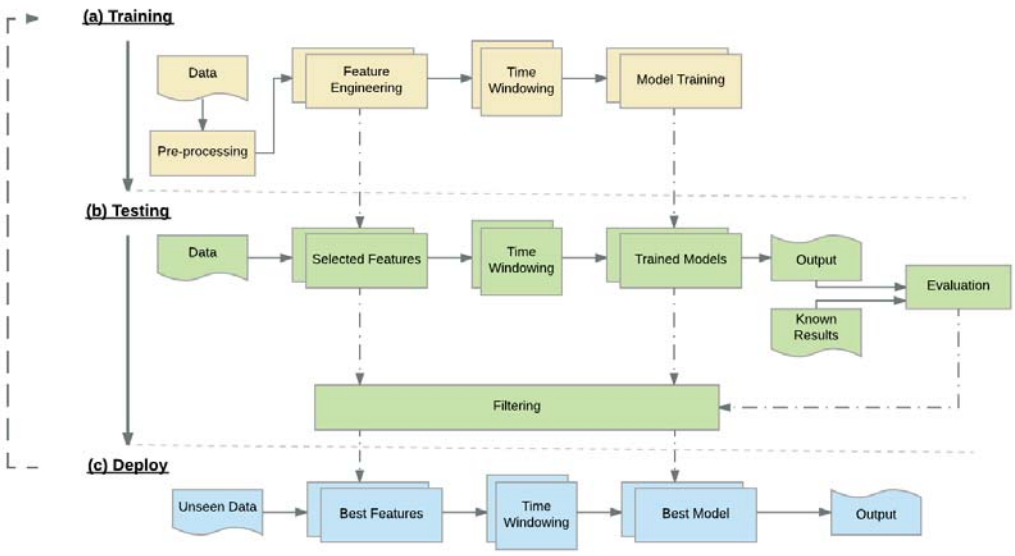

Figure 4. *Iterative process of applying and constructing ML-based prediction models.*

Objectives and Challenges

The primary objective of this study is to accurately predict water loss in KM water stations. The prediction is aimed for one-step and multi steps ahead in the future. The goal for the prediction results are to overcome current manual and baseline process to detect the water loss. Various techniques are to be investigated in order to achieve the purpose of constructing accurate prediction models. Specifically, the following are the work objectives and challenges of the thesis:

1. Identify and apply the proper data preprocessing steps prior to developing the prediction models.
2. Engineer effective features to be used to train and test the forecasting models.

3. Tune the developed models in order to achieve the highest accuracy for different horizons.
4. While working towards the previous objectives, we introduce and investigate new methods that yield improvements in performance.

Contributions

As it will be elaborated in the following chapters of the thesis, we have achieved and made the following contributions:

1. We built an integral multi-stage learning techniques that achieved the highest water loss prediction accuracy reaching 84.78 %.
2. Models are dynamically adaptable to data changes overtime.
3. Serve for different applications in water loss in different areas in water network.
4. Apply prediction for one step and multistep.
5. Feature modeling multivariate.

Thesis Organization

The rest of this thesis is organized as follows: background and literature review on related work is given in Chapter 2. Chapter 3 discusses the thesis methodology. In Chapter 4, we present and analyze the experimental results. Finally, we conclude the thesis and provide recommendations in Chapter 5.

CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

Current KM Water Loss Calculations

A series of challenges has recently emerged in water distribution field, triggered by the rapid shift in constructing and deploying enormous water reservoir utilities in order to satisfy customer's need. As one of the biggest challenges to comply with these changes is detecting water loss along the water network. The water SCADA datacenter in KM is responsible for calculating, detecting, and predicting water loss along the water network by continually collecting the distributed flow coming out from the desalination plants, reservoir and pumping station (RPS) inlet and outlet flow and pressure, reservoir levels and the DMA flow. Currently in KM, the collected data from the SCADA system is used to manually calculate the water loss estimation and prediction using mathematical formulas formed according to engineer's knowledge and experience.

Concretely, calculating the water loss is done in three parts, where the first part is computing the water loss from the desalination plants to RPS using equation (1):

$$water\_loss = \quad :5; \ L \ \hat{} \ \%@5 \ :\& \ A \ O = H \ EJ4 \leftarrow KP \ NE \ SK \ M(\ HKA \ S@F$$

$$\hat{} \ \breve{} \quad @5 \ :42 \ 5 \ JH \ 4A(\ HPK \ ;S \qquad (1),$$

Where D is the number of desalination plants, RPS is the number of RPS and the final result is in m^3.

The second part is to find the leaks inside the RPS station. This can indicate the usability and lifetime of the mechanical devices of the station such as pumps, valves and transmitters because the longer the age of the device, the higher possibility of leakage. This

part is divided into three steps. First, calculating the water stock of the RPS as in equation (2):

$$Water\_Stock = [\sum_{R=1}^{n}(Reservoir\_Level * Reservoir\_Capacity)_n] * 100 \quad (2),$$

Where R is the number of the reservoir inside the station and the final result in m^3.

The second step is to find the water loss by taking today's total stock, yesterday's total stock, RPS inlet total flow, and RPS outlet total flow into consideration when calculating the loss:

$$RPS\_Water\_Loss\ (m^3) = Today's\_Total\_Stock - (RPS\_Inlet\_Total\_Flow - RPS\_Outlet\_Total\_Flow) - Yesterday's\_Total\_Stock \quad (3),$$

The final step is getting the total water loss of all the stations, i.e., the network:

$$Water\_Loss_{part(3)} = \sum_{FM=1}^{n}(RPS\_Outlet\_Flow)_n - \sum_{DMA=1}^{m}(DMA\_Flow)_m \quad (4),$$

Where FM is the number of outlet flow meter inside the station, DMA is the number of district meter areas that the station pumping water to it and the final result in m^3.

There are many challenges encounters the engineers for calculating the water loss, which are:

- Time consuming because these manual calculations are done in daily, monthly and yearly basis.
- Cost ineffective because there are dedicated personnel that the company hires to do this specific task which cost the company financially.
- Data coming from SCADA is inaccurate at times or has zeros because of some error in the system. In these cases, they are ignored or not considering in the calculations

which will affect the results. Currently, there are no algorithms or models to follow in order to analyze these cases.

In this study, we will apply a data mining techniques to estimate and predict the water loss in water reservoirs to overcome the shortcomings of the existing manual processes.

Literature Review

In this section, we will look into different techniques used to analyze and calculate the water loss. These techniques can be classified into three categorize which are: analyzing and calculating water loss using machine learning, statistical, and hyper model approaches.

Statistical Approach

The proposed approach in Candelieri et al. (2014) merged both the machine learning methods and statistical hydraulic simulation of leakage. In this paper, Candelieri et al. developed a scenario (similarity) graph whose edges are weighted by the similarity between each pair of nodes. The similarity in terms of pressure and flow variations induced by two different leaks. This scenario will grant to move from feature space into scenarios Network Space and clarify the grouping problem of similar leakage scenarios as a graph clustering task. Their aim with clustering graph to maximize the sum of weights of edges within each cluster (intra-cluster similarity), at the same time minimizing the sum of the weights of the edges connecting nodes in different clusters (intercluster similarity) (Candelieri, Conti & Archetti, 2014).

The clustering method is used based on the eigenvalues analysis that is different

than the traditional clustering methods such as k-means. The proposed clustering method is called “spectral clustering,” that can be implemented by two techniques. The first technique is recursive bi-partitioning that divides the scenarios graph into two subgraphs and then is recursively used on each sub-graph until the wanted number of groups is achieved. The second technique is using the k-mean in the space identified by the smallest appropriate eigenvectors of the normalized Laplacian affinity matrix of the scenarios graph (Candelieri, Conti & Archetti, 2014).

The evaluation of the leak localization compared against the “localization Index”. The experiment is done in real WDN in a small town in the north of Italy by using the proposed k-means clustering (spectral clustering). At the same time, the dataset used in clustering is also employed by the regression model to estimate the prediction severity of the leakage. The regression model used is a simple least median squared linear regression which the researchers proved that is reliable by getting the Relative Mean Absolute Error= 0.8764% and Root Relative Mean Squared Errors= 2.5368% on 10 fold cross validation (Candelieri, Conti & Archetti, 2014).

Lijuan et al. (2012) reported a model-based leak detection method performed by the hydraulic simulation software of EPANET and optimized by the genetic algorithm. This process did not require extensive measurements or high capital cost but making use of a few monitoring pressure heads obtained from the water distribution networks.

The optimal value of the parameters will be drawn according to the objective function and constraint variables by compiling M-file in genetic algorithm toolbox (GUI) of Matlab and

setting running parameters (Lijuan, Hongwei, & Hui, 2012).

The genetic algorithm continuously updates a population of individual solutions. In each step, the genetic algorithm picks a random individual from the current population to be parents and consumes them to produce the children for the next generation (Lijuan, Hongwei, & Hui, 2012).

To extend the optimization toolbox capabilities, Matlab environment were used to solve the genetic algorithm. All the functions used are MATLAB M-files, made up of MATLAB statements that implement specialized optimization algorithms (Lijuan, Hongwei, & Hui, 2012).

The whole approach is applied to a test network; it comprises of 82 pipe sections with 79 pipelines in distributing the network, 53 junction nodes and 30 loops.

The estimated leakage amounts are smaller than the actual that is because that model built was based on virtual leak pipes and simulated leakage amounts that ignored the effects of leakage on water consumption and Supply of water (Lijuan, Hongwei, & Hui, 2012).

The weakness of statistical leak detection is that noise interferes in the statistical analyzes, and some leaks were hidden in the noise that prevented them from being detected.

Machine Learning Approach

In water networks, whenever there is a leak in the pipeline, the pressure of the flow drops suddenly in leak position and generates a negative pressure wave (NPW) in the pipeline. The pressure data can be collected by the pressure sensors installed in the pipeline,

and the negative pressure wave can be sensed by these sensors. Locating the leak can be determined by calculating the time difference between the arrival times of the negative wave in each sensor (Hou & Zhang, 2013). To analyze pressure readings a data mining approach is adopted to take a decision on the presence of the leak. Support vector machine (SVM) learning algorithm (Chen, Ye, Chen & Su, 2004) is used in this study. The NPW detection was performed as a two-class pattern classification task. The two classes are "NPW absent" and "NPW present". Along with SVM module, a nonlinear classifier is trained using supervised learning to detect the presence of NPW in pressure curve automatically. Thus, a small leak may be easily detected out of the noise. Furthermore, to process the pressure signal, a wavelet transform is used. Li Yo Bi (2010) also tested the wavelet transform for leak detection. The monitoring system acquired internal parameters of the pipeline from the existing SCADA system. The reported horizon for leak detection by this method is 2 minutes, and estimation error for leak localization is stated to be 2%.

Salam (2015), used a detector device to detect a leak and connected to a computerized system to collect and process the data by using SVM. As an application is implemented in Water Pipe Network System in Taman Khayangan Resident Makassar where an EPANET 2.0 simulation software is used to process the data. The data collected from the pipeline system is acquired from local drinking water company in Makassar. Then, this data is input into EPANET 2.0 to apply the SVM model to detect the location and size of the leak on the pipe. It was claimed that the results were accurate in predicting the leak size with root mean squared error (RMSE) average 0.06785 and leak location with

RMSE of 0.1382. Also, the average accuracy found for this application is 85.68% to predict the leak size and 76.14% to predict the leak location (Salam, 2015).

Nasir et al. (2014) use differential sensors of pressure that recognize a small change in the size of the leak. The water distribution system of Saudi Arabia is simulated and modeled in ERPANET application, and the input data is taken from this application. Then, DTREG and MATLAB software are used to process these data using ANN and SVM models. The conclusion of this experiment is that ANN model is more sensitive and not stable to noise than SVM. But, the performance of ANN is better if the noise is small.

The final results of the experiments were the value of the squared correlation coefficient (R) at errors of order 0.01 is 0.89 for SVM and 0.55 for ANN, and also the mean squared error (MSE) value is 1.2 for SVM and 12 for ANN. So, for their simulated system SVM model is more applicable for leak detection in a noisy environment than ANN models (Nasir, Mysorewala, Cheded, Siddiqui & Sabih, 2014).

On the other hand, by combining the machine learning approach to hydraulic simulation is what it is presented by Candelieri et al. (2014) paper. The goal was to enhance the leakage management by using an analytical leak localization by reducing costs and time for examining and repairing of water distribution network.

Many clustering algorithms are applicable; all of them require a particular measure (similarity or distance) to be decided in order to compare two objects that, in this case, are two vectors of flow and pressure difference at pipes and junctions. At the end of clustering process, a measure should be considered to allow the high quality of the solution with

respect to the goal (Candelieri, Soldi, Conti & Archetti, 2014).

An SVM model classifier has been used to train the data of the difference in flow and pressure of each scenario as input and the cluster supported by Spectral Clustering as target output (class label). So, the SVM classifier learns to approximate the non-linear mapping carried out by Spectral Clustering and to approximate the most probable cluster which the actual vectors of differences in flow and pressure belongs to (Candelieri, Soldi, Conti & Archetti, 2014). The best SVM classifier configuration settings used for this data are $C = 1$ and $\gamma = 1$. The learned SVM classifier has been validated on a separate test set, related to leakage scenarios retrieved on values of severity different from those already adopted, for example, new leaks. The study has been applied in a real case study, in two pilots of the European project ICeWater and Milan, Italy (Candelieri, Soldi, Conti & Archetti, 2014).

Herrera et al. (2010) compared and described multiple predictive models for forecasting water demand. The models are acquired using time series flow data from water consumption in an urban area of a city in south-eastern Spain.

Namely, the study took into account ANN, multivariate adaptive regression splines (MARS), projection pursuit regression (PPR), support vector (SVR) and random forests (RF) regression. Also, the researchers propose a simple model based on the weighted demand profile resulting from their exploratory analysis of the data (Herrera, Torgo, Izquierdo & Pérez-García, 2010).

The results of this comparison have recognized SVR models as the most accurate

models, approximately followed by MARS, PPR and RF. The experiments have also showed a weak performance of the different of neural networks that were considered. Finally, a heuristic model based on the empirical analysis of the regularities of the time series has shown its limitations when compared to these more sophisticated modelling approaches (Herrera, Torgo, Izquierdo & Pérez-García, 2010).

In one study by Kim et al. (2015) an effective daily water demand forecast method is used where two algorithms are combined: Genetic and neural network algorithms and they call it a neuro-genetic algorithm. The main feature of this study is adding more parameters that affect the water demand such as previous day's water demand, temperature, sunshine duration and day type that have a major impact on forecasting to give a better results. The experiment done in this study is for City of Seoul, South Korea.

The neural network model used in this experiment had a backpropagation algorithm in it and named in this study as NNBP and the genetic algorithm along with neural network model tested is named NNGA. Nine groups of NNBP are tested and trained. Then, nine groups of NNGA are created to get the best results in the water demand forecasting. The final results showed that applying neuro-genetic model with input parameters of two previous day's demand and todays along with the temperature parameter only give better performance forecasting today's water demand. The criteria used to judge if the performance is good or not are RMSE, Relative Root Mean Square Error, Absolute Mean Bias, and Mean Absolute Percentage Error. Out of 18 models that are tested, NNGA model found it has the best result where RMSE = 50422.33 (Kim, Hwang & Shin, 2011).

Hyper Approach

In Bakker's et al. (2014) Paper three various forecasting models for flow are studied: a transfer/ noise, an Adaptive Heuristic and multiple linear regression models. The performance of the models was analyzed both with and without using weather input, to determine the possible performance improvement because of using weather data. The final results showed that when using the weather data the largest forecasting errors may reduce by 11% and the average errors by 7%.

CHAPTER 3: METHODOLOGY

In this study, different models are built by learning from data. After the learning stage of the dataset, these models will be used to classify new data instances. The approach follows a process which is identical to the process introduced in knowledge discovery from databases (KDD) by Han and Kamber (2011). The process cycle and their order are described in *figure* 5 and outlined below:

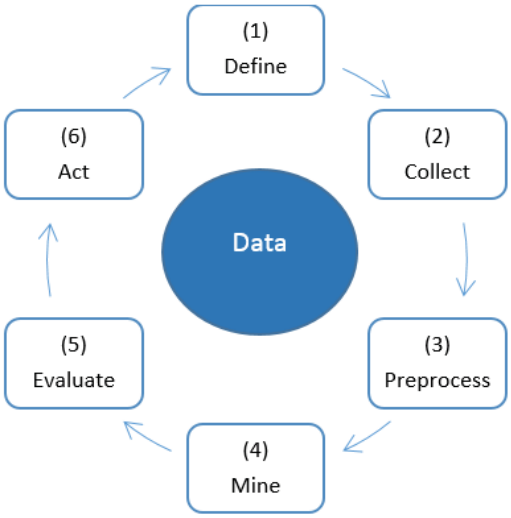

Figure 5. *The knowledge discovery process.*

1. **Defining Objectives:** In this cycle, the objectives of the exercised process are established and determined. One of these objectives could be model training and testing which consists of regression and classification models. Regression models are

one of the forecasting modeling technique where the target variable to be approximated is continued. On the other hand, classification is also one of the prediction modeling technique that predicts the class label of unknown records, so, the target label is discrete.

2. **Collecting Data:** This activity includes collecting of raw data and defining different features. To target this, data sources are recognized, and the features are chosen according to the objectives established in previous activity step and the subject area knowledge.
3. **Data Preprocessing:** This step evolves data cleaning from outliers and noise, handling missing values …etc. Data visualization and exploration are another techniques which are used to discover interesting trends and relations between the various features. Additionally, features or/and which are reduced and transformed for an efficient and effective performance of the following activity.
4. **Data Mining:** In this step, algorithms are applied, and models are built to accomplish the objectives stated in step 1. There exists a plenty of these algorithms to be exploited, and each could behave differently to the type of data and selected features. Additionally, some algorithms have parameters that sometimes need to be optimized to enhance the performance.
5. **Evaluation:** The output of the previous step is analyzed, interpreted and evaluated. Reliable statistical evaluations are usually used. For example, classifiers are evaluated by adopting a portion of the available data. It is used as hidden data that

have known classes; therefore, classifier output accuracy can be estimated. Additionally, the size of the hidden data is decided in relation to the over-all size of available data.

6. **Taking Actions:** If the evaluation is acceptable, models are applied. This is an application dependent step, which depends on previous steps. In classification, the best-selected features, and the presented classification models are used to categorize new data instance.

In a real application, these process steps are repeatedly performed whether partially or as a whole process. For example, in step 4, the algorithms are usually repeatedly tested and tuned. On the other hand, in step 5, to test the efficiency and effectiveness of the process. The details of the process activities related to this study are presented in the following sections (Benhmed, Shaban & El-Hag, 2014).

Data Collection

SCADA system which is implemented in KM collects sensory data such as reservoirs' inlet and outlet flows, reservoirs' levels, reservoirs' inlet water stock as well as the status of valves and used pumps in all over Qatar. These time-stamped data are periodically transmitted to several back-end servers for logging, storing, and processing. KM water network is composed of 35 connected stations; each includes from 3 to 10 reservoirs.

Feature Engineering

Data Pre-Processing

Data preprocessing is a wide field in data mining and contains a number of various techniques and strategies that are interrelated in different ways. The aim of data pre-processing is to boost the data mining analysis with respect to cost, time, and quality (Tan, 2006).

Data preprocessing consists of different techniques such as data transformation, data cleaning and data reduction. *Data transformation* is when the data transformed to give a more efficient data mining result and the data patterns is easier to understand. Two forms of transformation are mostly used discretization and normalization. *Data cleaning* goal is to solve data inconsistency, missing values and reduce data outliers. *Data reduction* goal is to reduce the representation of the dataset while obtaining integrity of the original data and give more efficient analytical result (Han, Pei & Kamber, 2011).

Feature Selection

Feature Selection is one of the ways to determine which attribute features to be input it to the model. This technique reduces the dimensionality of the data to enhance the performance of the learning algorithm, easiness in interpreting the data and speed up the learning (Hall, Witten & Frank, 2011).

The feature selection can be *heuristically* through a well understanding of the data patterns and features, or *automatically* by using filtering methods or wrapper methods (Tan, 2006).

In our study, we use the heuristically method for feature selection of the dataset by relying on KAHRAMAA's operation expertise for calculating the water loss and by automatically by using decision tree models such as J48, RF and Random trees(RT).

Time Windowing

Windowing is one of the data mining transformation technique mostly used in prediction of time series data. Since KM's dataset is a time series data and our target to predict the water RPS loss. In general, time series data and its transformed structure are conceptually shown below in *figure 6*:

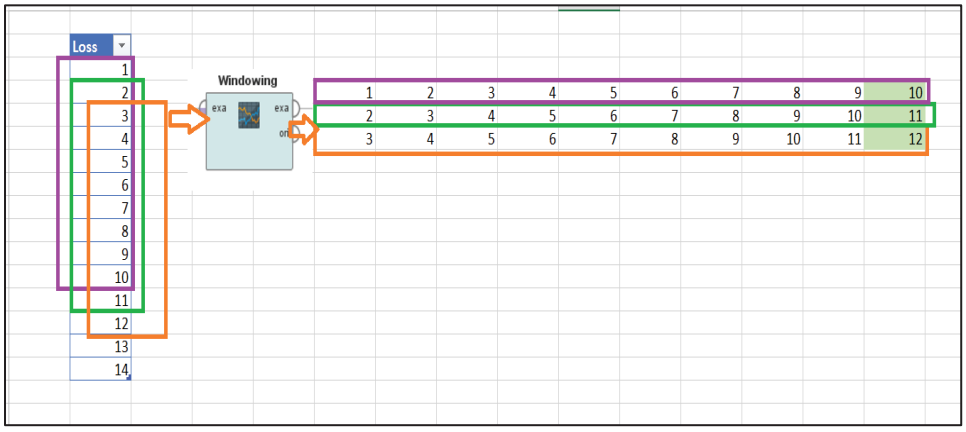

Figure 6. *Windowing technique.*

The parameters of the "Windowing" operator grant changing the size of the windows as shown in *figure 6* the colored vertical boxes on the left. While the overlap between the windows is known as a step size and the last value in a window is called a

label variable (green box in the figure above) which is applied for forecasting. Thus a time series data is now changed into a generic data set which can be processed by any of the available data mining operators.

Model Training and Testing

Regression and classification models are machine learning methods for building forecasting models from dataset given. The models are acquired by recursively partitioning the dataset and fitting a simple forecasting model within each partition. Accordingly, the partitioning can be displayed graphically as a decision tree. Regression trees are constructed for dependent variables that take ordered discrete values or continues and predict a particular value in future. On the other hand, Classification trees are constructed for dependent variables that take an unlimited number of unordered values and predict a range in future.

Regression

Linear Regression

Linear regression (Han, Pei & Kamber, 2011), uses the equation of a straight line which is $y = mx + b$ and figures out the suitable values for m and b to predict the value of y depends on value of x. Mostly, the Linear regression models are used to show the relationship between two factors. The factor that is being forecasted is called the dependent variable. The factors that are used to forecast the value of the dependent variable are named the independent variables.

Multilayer Perceptron (MLP)

MLP is a static neural system that contains consecutive layers which exchange and communicate the information through synaptic connections expressed by an adaptive weight. The MLP structure comprises an input layer which is built of some perceptions equal to the number of data attributes. On the other hand, the output layer contains one perceptron in regression or more when the task for classification, so the number of the perceptron is identical to the number of classes to be predicted, while rest of the other layers are hidden (Tan, 2006).

Sequential Minimal Optimization Regression (SMOReg)

The Sequential Minimal Optimization (SMO) algorithm was created by John Platt in 1998 which is usually combined with Radial Basis Function (EBF) kernel used for prediction process. In WEKA program, SMO algorithm uses SVM for learning and the model named SMOReg (Shevade, Keerthi, Bhattacharyya & Murthy, 2000).

Classification

J48

J48 is one of the decisions trees that applies Quinlan's C4.5 algorithm (1993) for producing unpruned or pruned C4.5 tree. Moreover, the C4.5 algorithm is an improvement of Quinlan's ID3 algorithm. The decision trees produced by J48 usually used for classification, where J48 algorithm forms decision trees from a group of labeled training data applying the concept of information entropy. It applies the case that every attribute of the data could be used to create a decision by breaking the data into smaller subsets.

J48 analyzes the normalized information gain (difference in entropy) that results from selecting an attribute for dividing the data. To build or give a decision, the highest normalized information gain attribute is chosen. Next, the algorithm repeated on smaller subsets. The stop condition of splitting reached if all instances in a subset associated with the same class. After that, a leaf node is built in decision tree informing to select that class. However, it can also occur that none of the features deliver any information gain. In that case, J48 algorithm constructs a decision node higher up in the tree employing expected value of the class (Quinlan, 1993). The pseudo code of J48 shown below in *figure 7*:

1. Check for base cases.
2. For each attribute a:
    a) Find the feature that best divides the training data such as information gain from splitting on a.
3. Let a_{best} be the attribute with highest normalized information gain.
    a) Create a decision node that splits on a_{best}
4. Recurs on the sub-lists obtained by splitting on a_{best} and add those nodes as children of node. Stop when the stopping condition is met

Figure 7. *The pseudo code of J48 algorithm (Quinlan, 1993).*

J48 algorithm works on both discrete and continuous attributes, attributes with distinct costs, and training data with missing attribute values. Also, it affords an option for pruning trees after creation (Quinlan, 1993).

RF

RF is a novel algorithm to the area of data mining and is constructed to generate accurate predictions that do not over-fit the data (Breiman, 2001). RF algorithm builds several trees, and each tree is created with randomized subset predictors. Therefore, the name "random" is called. An enormous number of trees (500 to 2000) developed, so, "forests" of trees is called. The number of predictors applied to discover the best split at every node is a randomly picked subset of the total number of predictors, the trees are getting larger to a maximum size without pruning, and aggregation is done by averaging the trees. Out of the bag samples usually applied to compute an unbiased error rate and variable importance, removing the need for a test set. On the other hand, a big number of trees are built, there is a restricted generalization error that means that no overfitting is attainable which is an advantage for forecasting.

By building each tree to maximum size without pruning and picking only the best split of the random subset at each node, RF algorithm attempts to control some prediction strength while producing a distinction between trees (Breiman, 2001). Random predictor selection reduces the correlation between un-pruned trees and maintains the bias low by handling an ensemble of unpruned trees; variance is also reduced. The pseudo code shown in below *figure 8* summarized the RF Algorithm. One of the advantages of RF algorithm

is that the forecasted output relies on only one user-selected parameter. Hence, the number of predictors to be picked is random at each node.

Let N_{trees} be the number of trees to build for each of N_{trees} iterations.

1. Select a new bootstrap sample from training set.
2. Grow an un-pruned tree on this bootstrap.
3. At each internal node, randomly select m_{try} predictors and determine the best split using only these predictors.
4. Do Not perform cost complexity pruning. Save tree as is, alongside those built thus far.

Output overall prediction as the average response (regression) or majority vote (classification) from all individually trained trees.

Figure 8. *The pseudo code of RF algorithm (Breiman, 2001).*

RT

RT models have been widely used in the area of machine learning recently. When we have k random features at each node, a RT is a tree form at random from a group of possible trees. Therefore, the name "random" means that each tree in the group of trees has

an equally chance of being sampled. So, the division of trees is "uniform." RT could be created efficiently, and the mixture of large groups of RT mainly leads to an accurate model (Han, Pei & Kamber, 2011).

Ensemble Methods

This section introduces some methods for enhancing classification accuracy by collecting the forecasts of multiple classifiers. These methods are well-known as ensemble methods. The ensemble method builds a group of base classifiers from training data and carries out a classification models by taking a vote on the forecasts made by each base classifier. This section clarifies why ensemble methods likely to carry out better than any single classifier.

Boosting

Boosting (Han, Pei & Kamber, 2011) is a repetitive procedure used to alter the distribution of training examples so that the base classifiers will spotlight examples that are complicated to classify. Dissimilar than bagging, boosting appoints a weight to each training example and could frequently change the weight at the end of each boosting cycle. The weights appointed to the training examples may be used in the following ways:

1. The weights may be used as a sampling distribution to build a group of bootstrap samples from the original data.
2. The weights may be used by the base classifier to learn a model that is biased to a higher weight examples.

In past years, many implementations of the boosting algorithm have grown in

different fields. These algorithms distinct in terms of:

1. How the weight of the training examples are amended at the end of boosting cycle.
2. How the forecasting builds by each classifier are linked.

An implementation called AdaBoost algorithm investigated next.

AdaBoost. Allow $\{(\mathrm{x_j}, \mathrm{y_i})|\mathrm{j} = 1,2, \ldots, \mathrm{N}\}$ express a set of N training examples. As primary process of AdaBoost algorithm, the priority of a base classifier $\mathrm{C_i}$ counts on its error rate, which is expressed in Equation 5 as

$$\epsilon_i = \frac{1}{N}\left[\sum_{j=1}^{N} \omega_j \, I\big(C_i(x_j) \neq y_j\big)\right]$$

Equation 5: Counting error rate in AdaBoost algorithm

Where $I(P) = 1$ if the forecast P is true, otherwise 0. The significance of classifier C_i is defined by the flowing parameter in Equation 6,

$$\alpha_i = \frac{1}{2} ln\left(\frac{1 - \epsilon_i}{\epsilon_i}\right)$$

Equation 6: The importance of classifier C_i

Hence that if the error rate is close to 0, α_i has a large positive value and if the error rate close to 1, α_i has a significant negative value.

The parameter α_i is also applied to update the weight of training examples. To demonstrate, let $\omega_i^{(j)}$ express the weight assigned to example (x_i, y_i) during the j^{th} boosting cycle. The weight update method for AdaBoost algorithm is represented in Equation 7:

$$\omega_i^{(j+1)} = \frac{\omega_i^{(j)}}{Z_j} \times \begin{cases} exp^{-\alpha_j} & if\ C_j(x_i) = y_i \\ exp^{\alpha_j} & if\ C_j(x_i) \neq y_i \end{cases}$$

Equation 7: The weight update method in AdaBoost algorithm

Where Z_j is the normalization factor applied to guarantee that $\sum_i \omega_i^{(j+1)} = 1$. The weight update equation which is expressed in Equation 7 decreases the weight of examples classified correctly and increases the weights of incorrectly classified examples.

Rather than using a majority voting scheme, the forecasting made by each classifier C_i is weighted depending on α_i. This method grants AdaBoost algorithm to punish models that have low accuracy. Furthermore, if any in-between cycles give an error rate bigger than 50%, the weight are returned to their original uniform values, $\omega_i = 1/N$, and the resampling method is done again (Freund & Schapire, 1996).

Bagging

Bagging or bootstrap aggregating, is an approach that frequently samples (with replacement) from a dataset conforming to a uniform probability distribution. Each bagging sample has an identical size as the original data. By reason, the sampling is done with replacement, part of instances usually presents many times in the same training set, while others could be excluded from the training set. On average, a bagging sample D_i consists nearly 63% of the original training data due to every sample has a probability $1 - (1 - \frac{1}{N})^N$ of being chosen in each D_i . If N is adequately large, that probability converges to$1 - 1/e \cong 0.632$. After training the k classifiers, a test instance elected to the class that earns the highest number of votes (Breiman, 1996).

Comparing Experimental Results

Cross Validation

In cross-validation, each record in the dataset is used the same number of iterations for training and testing. To demonstrate this method, suppose we divide the data into two equal-sized subsets. First, we select one of the subsets for testing and the other is for training. Then, we exchange the roles of the subsets so that the previous testing dataset becomes the training dataset and vice versa. This method is called a two-fold cross-validation. On the other hand, the k-fold cross-validation approach generalizes the cross-validation method by dividing the data into k equal-sized partitions. In each run, one of the partitions is selected for testing while the rest of them selected for training. The process is repeated k times so that each partition is selected for testing exactly once.

Cross-validation method has the advantage of applying as much data as possible for training. Also, the test datasets are mutually exclusive, and they effectively cover the entire dataset.

Baseline

The RMSE and accuracy rate is not very meaningful to show the forecasting performance of the model's algorithm used in this experiment. Therefore, in this experiment, we compare the classification models results against a baseline model results to show how the data mining algorithm models enhance the forecasting concept to detect the water loss in water stations.

The simplest models used in this experiment is to take the dataset as a baseline and

discretize the attribute (RPS LOSS) into three and four bins with equal frequencies, then blindly classify each value with its significant bin. The final step, we calculate the correctly classified instances in baseline and compare it with correctly classified instances in each classification models to emphasize the prediction performance of classification models against the baseline.

CHAPTER 4: EXPERIMENTAL RESULTS

Dataset Collection

In this study, Airport water station hourly data has been chosen to apply the data mining algorithms and analyze it. The data used are:

1. Water Station Inlets: It is a flow meter reading that reads how much water entering the station from different desalination plants. The readings in Million Gallon (MIG).
2. Water Station Outlets: It is a flow meter reading that reads how much water pumping out the station to different district areas. The readings in Million Gallon (MIG).
3. Water Reservoir level: It is the water level in each tank inside the station. The readings in Meter (m).

Data Preprocessing

There are some negative and zero data has been presented in the dataset which is mostly presented in flow meters readings for the inlet and output data of the station and in reservoir level data. These noisy data are presented in input and output flow meters data are because of communication loss, faulty device, or there is no water in the pipeline. The following solution has been done:

- if zeros presented consecutively (more than hour) and the station pumping water to customers means it is a faulty device (not recording the data) in this case we don't ignore these zeros we take average of last collected data with the first data taken after the zero data and replace the result in all zeros.
- If zeros presented consecutively (more than hour) and the station not pumping

water to customer means there is no water in the pipeline in this case we leave the zero as it is.

- If zeros presented only one-hour means there is communication loss or power disruption in this case we take the same action as the first case by taking the average.

On the other hand, noisy data also presented in reservoir tank level data which is because of various reasons but mostly because of Loss of communication, first construction of the reservoir tank or because there is a maintenance in the reservoir tank. The following solution has been applied to overcome this noisy data:

- If the zeros presented at the beginning and last more than a week means that the reservoir is in construction phase and not yet in service in this case the zeros are ignored or deleted.
- If the zeros presented for few hours and we see the zeros in all the reservoirs means there is loss in communication of the reservoir and in this case we don't ignore these zeros we take average of last collected data with the first data taken after the zero data and replace the result in all zeros.
- If the zeros presented for few hours in this particular reservoir means there is maintenance in the reservoir and in this case the zeros are ignored or deleted.

Feature Engineering

Formatting Data

An additional columns has been added to represents Qatar's temperature, season, holiday and day of the week. The season column is added to represents the weather seasons in Qatar, in order to be consistent and to make forecasting easier, meteorologists divide the year into 4 meteorological seasons of 3 months each (numbered accordingly):

1. Spring: starting from 1st of March 1 till 31st of May.
2. Summer: starting from 1st of June 1 till 31st of August.
3. Fall: starting from 1st of September till 30th of November 30.
4. Winter: starting December 1 and ending February 28 (February 29 in a Leap Year).

The "day of the week" column is added to let the system recognize and learn the data and recognize the water loss during the week days. The numbering of the week days is as the following:

1. Sunday (1)
2. Monday (2)
3. Tuesday (3)
4. Wednesday (4)
5. Thursday (5)
6. Friday (6)
7. Saturday (7)

On the other hand, the highest demand and losses also depends on the holiday

seasons, the Eid, school break and Ramadan seasons are mostly have the highest demands. Also, three columns are created which are the “Total Inlet” which sum up all the water inlets readings, “Total Outlets” which sum all the water outlet readings and “Total Stock” which represent the sum of all reservoir’s level. Moreover, an additional column is added “RPS LOSS” that represents the hourly loss in the station, equation 3 is used to calculate the loss in each hour.

The file used is in CSV format in order to analyze it via Rapid Miner application.

Feature Selection

According to KAHRAMAA’s operation for calculating the water loss, I have decided to use the following attributes that have a major influence on data mining algorithms:

- Month
- Day
- Day of the week
- Hour
- Total Inlet
- Total Outlet
- Total Stock
- RPSLOSS
- Temp

- Season
- Holiday

Snapshot of the dataset used is shown below in *figure 9*:

Month	Day	Day of the week	Hour	Total Inlet	Total Outlet	Total Stocl	Temp.	Season	Holiday	RPS_LOSS
11	17	1	12	3958.44	4692.06	90904.19	21	3	0	513.82522
11	17	1	13	3944.19	4688.82	90673.39	21	3	0	534.65742
11	17	1	14	3950.79	4677.48	90481.36	22	3	0	547.03651
11	17	1	15	3953.61	4715.28	90266.72	22	3	0	576.59954
11	17	1	16	3940.68	4733.64	90050.36	21	3	0	554.42193
11	17	1	17	3932.4	4735.8	89801.38	22	3	0	514.7352
11	17	1	18	3930.63	4696.38	89550.37	22	3	0	533.3483
11	17	1	19	3930.18	4723.92	89289.98	23	3	0	570.74684
11	17	1	20	3914.46	4703.94	89071.24	23	3	0	333.76384
11	17	1	21	3912.84	4411.8	88906.05	22	3	0	341.52463
11	17	1	22	3926.37	4318.92	88855.02	23	3	0	344.24259
11	17	1	23	3946.8	4236.3	88909.76	23	3	0	103.50938
11	18	2	0	3948.36	3898.26	89063.37	23	3	0	53.24738
11	18	2	1	3951.66	3628.26	89440.02	23	3	0	-91.7578
11	18	2	2	3951.78	3261.6	90038.44	24	3	0	63.73119
11	18	2	3	3940.44	3181.14	90861.47	24	3	0	162.49381
11	18	2	4	3957.33	3224.88	91756.42	24	3	0	598.98648
11	18	2	5	3935.28	3724.92	92565.76	24	3	0	784.37291

Figure 9. *Dataset snapshot.*

Therefore, the inlet, outlet and reservoir level columns are deleted since we sum them all.

Data Windowing

The main items in Windowing is the window size used which is 24 to represent a day, and the step size used in this experiment 12 because the water distribution pattern is changing every 12 hours. While, the horizon used is 1, 12 and 24 as recommended by

KAHRAMAA engineer expert. On the hand, to test the accuracy of daily dataset an hourly dataset is used to compare the accuracy and mean squared error between them.

Implementation and Tools

Two processes used in this experiment, the first process is used to test the classification models as shown in below *figure 10*, where four operators for setting up windowing and analyzing the predicted data. “Read CSV” operator is used to import the time-series data needed to be predicted which is saved as a CSV file. All time series will a date column (Month, Day, and Hour) and this must be treated with special care. The imported data is then windowed with “windowing” operator which is installed in Rapid Miner application via the series extension that windowed the data into different sizes, steps and horizons which make the prediction process of water loss more predictable and easier to learn. Also, it creates a new label attribute accordingly for the water loss. Then, the data is discretized by “Discretize by Frequency” operator which discretize the label attribute into different number of bins depending on the range of data, in this experiment the attribute is discretized into 3 and 4 bins for testing to predict the water loss and classify them according the bins number. The last operator is the “Cross Validation” operator which partitioned the data set into *k* subsets of equal size. Out of the *k* subsets, a single subset is retained as the testing data set and the remaining $k - 1$ subsets are used as training data set. The cross-validation process is then repeated *k* times, with each of the *k* subsets used exactly once as the testing data. The *k* results from the *k* iterations then combined to produce a single estimation. The value *k* is adjusted using the *number of validations*

parameter which is 10 iterations in our experiment. Hence that the second process used is the same as the first process except the discretization operator is not used for testing the regression algorithm models.

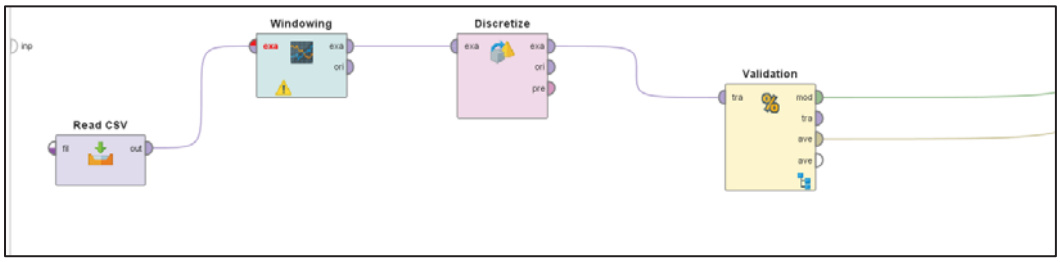

Figure 10. *Main process implementation.*

The validation process which is shown in *figure 11* has two sections, the training and testing sections where a WEKA learning classification and regression algorithm models are implemented in training section. On the other hand, the performance operator in testing section is used to produce accuracy and error results of the classification and regression prediction.

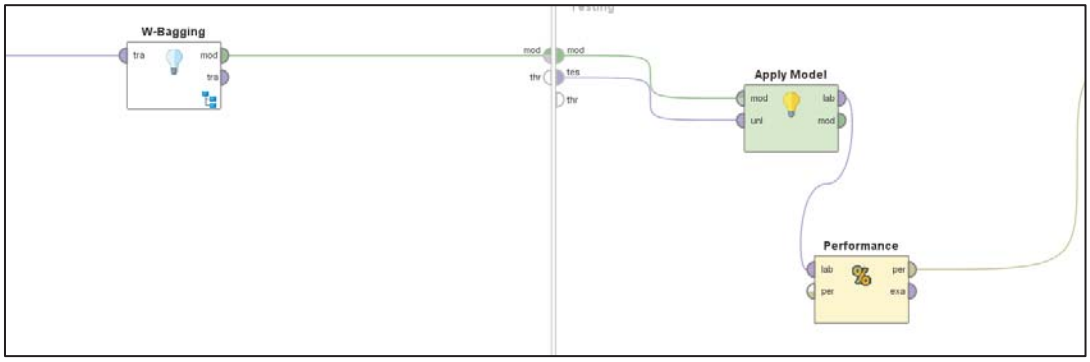

Figure 11. *Validation process environment.*

Forecasting Algorithms Setup

The algorithms used in batch procession is tuned to improve the prediction accuracy. Which is implemented in validation process for learning and predicting the data. The classification models algorithms settings is shown in table (1), the regression models algorithms settings is shown in table (2) and ensemble methods settings shown in table (3).

Table 1

Classification Model Algorithms Parameters

Algorithm	Parameter	Value	Justification
J48	confidence threshold for pruning	0.2	Pruning reduces the complexity of the final classifier, and improves predictive accuracy by the reduction of overfitting.
	Min. No. of instances per leaf	2	Depends on the dataset itself, so it is tuned to improve the accuracy of the result
RF	No. of trees to build	10	Depends on the dataset itself, so it is tuned to improve the accuracy of the result
	No. of features	0	This feature is when we need to manually set the number of trees. However, to consider an unlimited number of feature so we set it to 0.
	Max. Depth of trees	0	This feature is when we need to manually set the number of trees. However, to consider an unlimited number of feature so we set it to 0.
RT	No. of attributes randomly investigated	10	In our test we didn't tune this feature since we are testing it.
	Min. No. of instances per leaf	1	Depends on the dataset itself, so it is tuned to improve the accuracy of the result
	Seed for random number generator	1	The default number is one and it gave more accurate results.

Table 2

Regression Model Algorithms Parameters

Algorithm	Parameter	Value	Justification
Linear Regression	Attribute selection method	0	Depends on the datasets itself, so it is tuned to improve the accuracy of the result.
Multilayer Perceptron	Learning rate for the backpropagation algorithm	0.3	Depends on the datasets itself, so it is kept its default value.
	Momentum Rate for the backpropagation algorithm	0.2	Depends on the datasets itself, so it is kept its default value.
	Number of epochs to train through.	500	The number of trained epochs used depends on the dataset. In this case, the default value used.
SMOReg	The complexity constant	1	In our test we don't tuned this feature since we are testing it.
	Normalization	0	It means that we want to normalize the result.
	Seed for random number generator	1	The default number is one and it gave more accurate results.

Table 3

Ensemble Method Algorithms Parameters

Algorithm	Parameter	Value	Justification
AdaBoostM1	Percentage of weight mass to base training on	100%	The default number is 100 and it gave more accurate results.
	Number of iterations.	14	Depends on the datasets itself, so it is tuned to improve the accuracy of the result.
Bagging	Size of each bag	100%	As a default, the size of the training size is the whole dataset.
	Number of iterations.	14	Depends on the datasets itself, so it is tuned to improve the accuracy of the result.

Experimental Evaluation

Cross validation is tuned to validate the results of the tested models, *number of validations* parameter is set to 10 validations which specifies the number of subsets that the dataset should be divided into. Furthermore, the same number of iterations will take place, each iteration covers testing and training the model. *Sampling type* used is stratified sampling since there is a label attribute. Both accuracy and RMSE results discussed in chapter 3 are reported for all experiments.

On the other hand, the experiment's results are compared to a baseline results to show the prediction accuracy and errors.

Results

In this section, we represent all the results accomplished in regression and classification forecasting models. In Section 4.4.2.1, we illustrate the Regression models experiments using Linear Regression, Multilayer Perceptron, and SMOReg algorithms on windowed and non-windowed data. In Section 4.4.2.2, we show the Classification models experiments using J48, Random Forest, and Random Tree algorithms on windowed data. Also, in this section we show the experimental results of ensemble classification models using Boosting and bagging methods and compare the classifications results with the baseline results.

4.4.2.1 Regression

In this section, we need to study the effect of regression models on windowing to calculate prediction trend accuracy (PTA) and RMSE. Table 6 in the appendix shows the PTA percentage and RMSE for predicting the RPS water loss across horizon 1, 12 and 24 using Linear Regression, Multilayer Perceptron and SMOReg algorithms and the PTA percentage and RMSE value for non-windowed dataset.

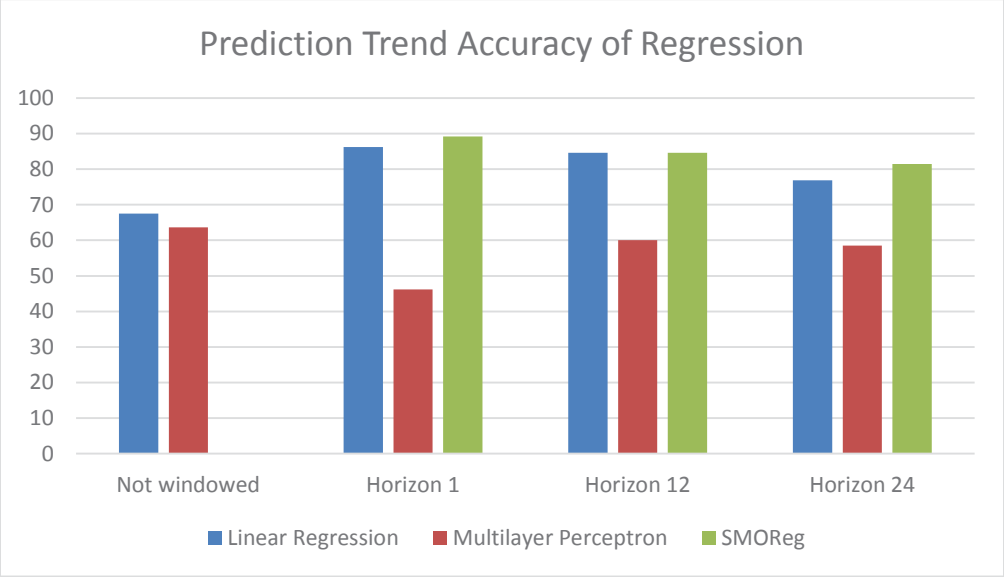

Figure 12. *PTA for regression models.*

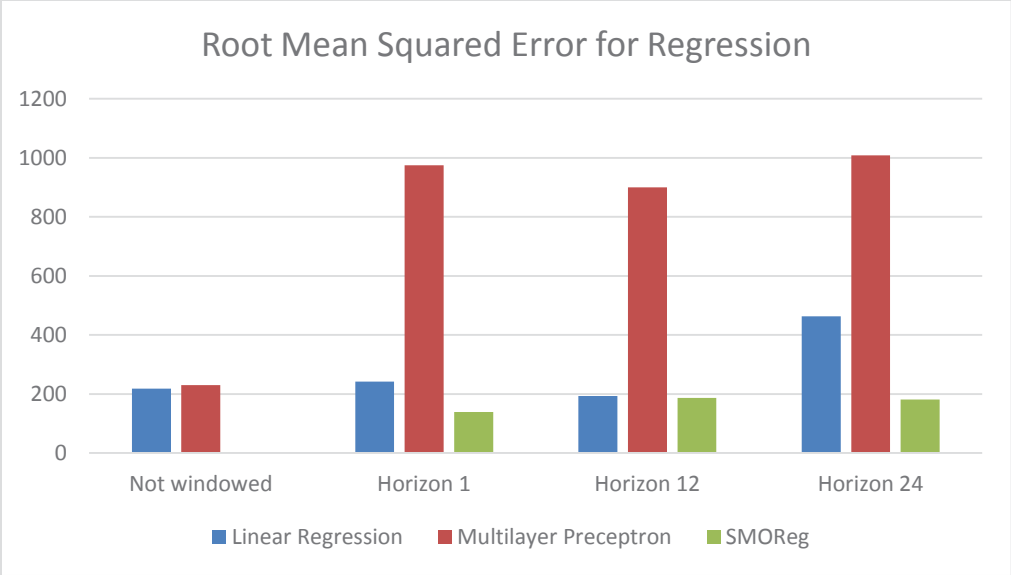

Figure 13. *RMSE for regression models.*

Figure 12 and *Figure 13* illustrate the best values of PTA percentage and RMSE values calculated. Each column in the figures has a label indicating the algorithm used to achieve this result. Based on figure, we can conclude the following:

- Not windowed dataset achieves less PTA compared to windowed data.
- SMOReg algorithm outperforms other algorithms across all horizons.
- PTA percentage for predicting RPS water loss is high by using Linear Regression and SMOReg algorithms across all horizons ranging between 80's and 90's.
- RMSE values for all algorithms is very high across all horizons ranging between 100 and 1000.

4.4.2.2 Classification

In this section, we need to study the effect of classification models on windowing and discretization of the dataset to calculate accuracy percentage and RMSE value. Table 7 and 8 in the appendix shows the accuracy percentage and RMSE value for predicting the RPS water loss across horizon 1, 12 and 24 using J48, Random Forest and Random Tree algorithms with standard deviation values for all algorithms. Also, in this section, we represent the ensemble classification methods (Boosting and Bagging) as results shown in Table 9, 10, 11 and 12 in the appendix. Finally, we need to validate the classification models results with the baseline results represented in Tables 13-18 in the appendix.

Table 4

Three Bins Classification for J48 Algorithm in Horizon 12.

	A	B	C
A [-∞- 410.455]	144	48	27
B [410.455 - 531.257]	51	98	67
C [531.257 - ∞]	25	75	127

Table 5

Four Bins Classification for Random Forest Algorithm in Horizon 24.

	A	B	C	D
A [-∞- 362.376]	113	35	16	11
B [362.376 – 473.306]	31	68	56	15
C [473.306- 577.658]	12	47	56	44
D [577.658- ∞]	9	15	37	96

Table 4 and Table 5 demonstrates an output screen shot of two confusion matrix results of two algorithms where it shows the ranges and how many correctly classified instances in each range. From these screenshots we conclude the following:

- Table 4 which shows a 3 bins classification of J48 algorithm in horizon 12, 144 instances correctly classified or predicted to be in range [-∞ - 410.455] which represent a very high loss range, 98 instances correctly classified or predicted to be in range [410.455 - 531.257] that represent a small water loss range and 127 instances correctly classified or predicted to be in range [531.257- +∞] which represents no loss range.
- Table 4 which shows a 4 bins classification of Random Forest algorithm in horizon 24, 113 instances correctly classified or predicted to be in range [-∞ - 362.376] which represent a very high loss range, 68 instances correctly classified or predicted to be in range [362.376 – 473.306] that represent a small water loss range, 56 instances correctly classified or predicted to be in range [473.306– 577.658] also represents a small water loss range and 96 instances correctly classified or predicted to be in range [577.658 - +∞] which represents no loss range.

The confusion matrix is presented in the appendix for all classification models across all horizons.

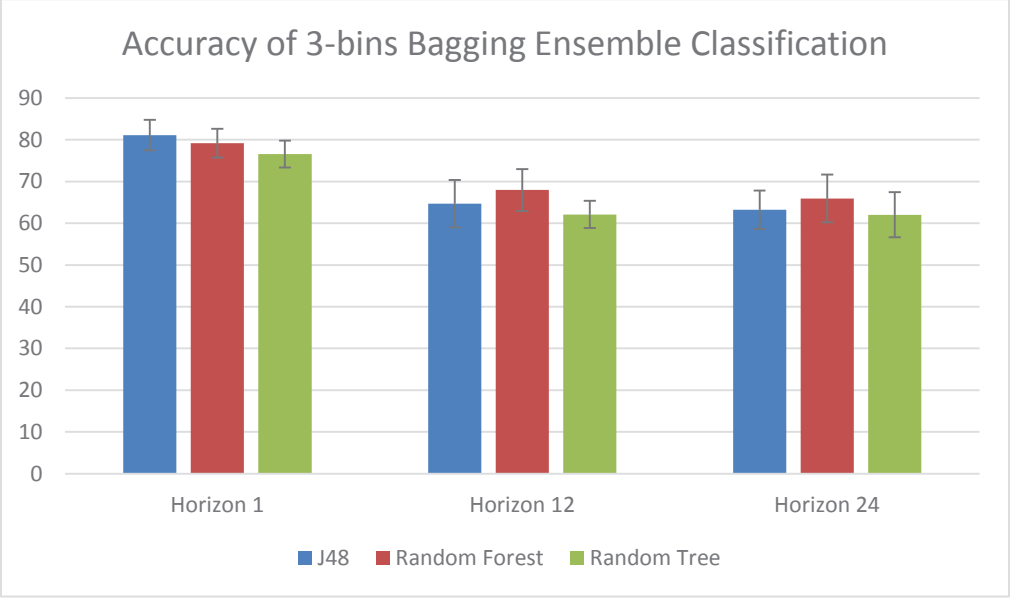

Figure 14. *Accuracy for 3-Bins discretization of bagging-ensemble classification models.*

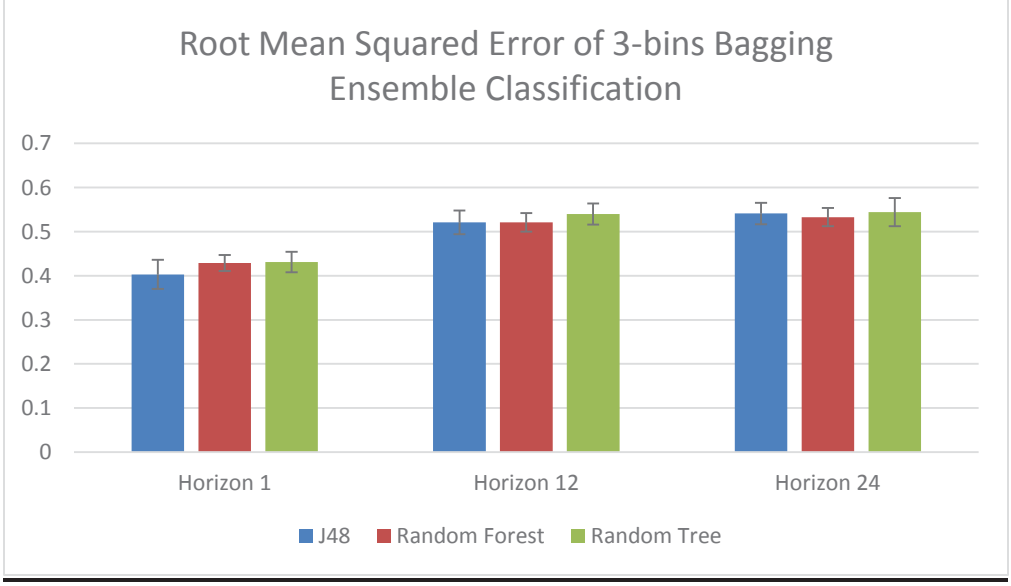

Figure 15. *RMSE for 3-bins discretization of bagging-ensemble classification models.*

Figure 14 and *Figure 15* illustrate the best values of accuracy percentage and RMSE of 3-Bins Discretization of Bagging-Ensemble Classification Models. Each column in the figures has a label indicating the algorithm used to reach this result, and the standard deviation value is also represented by each column as error bars. Based on the figures, we conclude the following:

- Random Forest algorithm performs best compared to other algorithms for predicting the water loss in horizon 12 and 24.
- Accuracy in horizon 1 which represents predicting the water loss after 1 hour is high for all algorithms which are reasonable since it is easy for the classifier to predict the next hour and getting complicated in predicting further hours. The water loss prediction accuracy for all algorithms in horizon 1 around 80's.
- RMSE is minimal in the first horizon and keeps increasing in other horizons. However, Random Forest and J48 algorithms have low RMSE values in horizons 12 and 24.

To validate the accuracy of the classification results of all models we experimented, we compare them with the baseline model results that is represented in Table 13 to Table 18 in the appendix for 3-bins and 4-bins discretization across horizons 1, 12 and 24.

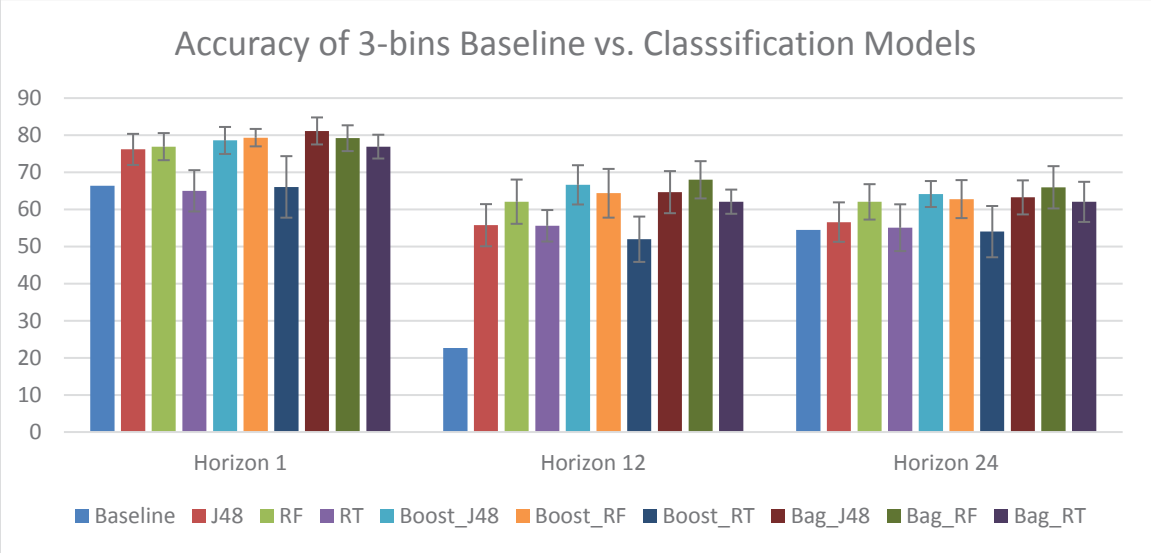

Figure 16. *Accuracy for 3-bins discretization of all classification & baseline model.*

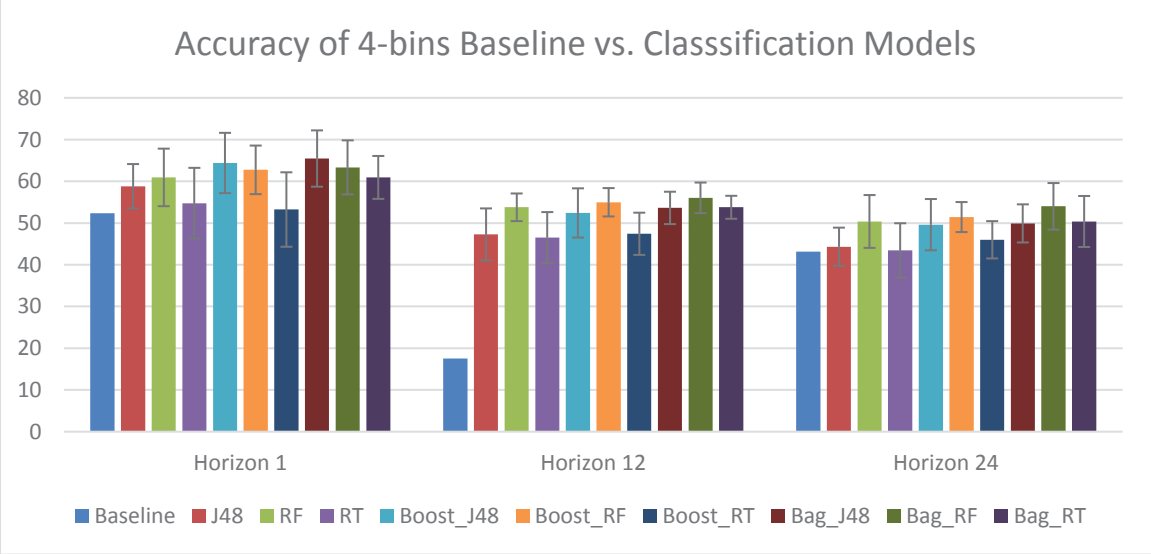

Figure 17. *Accuracy for 4-bins discretization of all classification & baseline model.*

Figure 16 and *Figure 17* illustrates the best achieved accuracy percentage for all Classification Models and Baseline Model. Each column in the figures indicate the algorithm used to achieve the results, RF for Random Forest, RT for Random Tree, Boost_J48 for J48 algorithm using Boosting-Ensemble classification, Boost_RF for Random Forest algorithm using Boosting-Ensemble classification, Boost_RT for Random Tree algorithm using Boosting-Ensemble classification, Bag_J48 for J48 algorithm using Bagging-Ensemble classification, Bag_RF for Random Forest algorithm using Bagging-Ensemble classification, and Bag_RT for Random Tree algorithm using Bagging-Ensemble classification. Based on both figures we can conclude the following:

- 3 bins discretization for all models performs better than 4 bins discretization for all models. The range of accuracy for all models in 3 bins discretization between 20% and 80% but in 3 bins discretization is between 19% and 70%
- The baseline model accuracy performs nearly the same as all classification models in horizon 1 in 3 bins and 4 bins discretization. On the other hand, baseline model accuracy performs worse in horizons 12 and 24 for both 3 bins and 4 bins discretization.
- Random Forest algorithm accuracy performs the best by using Bagging-Ensemble classification model compared to other algorithm models in horizon 12 and 24 for 3 bins and 4 bins discretization. Therefore, the prediction accuracy of water loss after 12 hours in 3 bins discretization around 73.01% and 4 bins discretization around 71.77%.

- The best accuracy outperformed in horizon 1 is J48 algorithm by using Bagging-Ensemble classification model in 3 bins discretization which gave around 84.78% accuracy.
- The standard deviation of Random Tree algorithm for all classification models has the highest values across all horizons in 3 bins and 4 bins discretization.

CHAPTER 5: CONCLUSION AND FUTURE WORK

After doing several experiments on water loss in one of the KAHRAMAA's water station using various machine learning techniques, we have exposed to several conclusions and future work guidelines that are discussed in details in the following sections.

Conclusion

Water loss is one of the major concern that effect water utilities especially in KAHRAMAA Corporation in Qatar. Water loss effect the environment, cost and resources that effects negatively on the corporation. These water losses need to be monitored, studies and analyzed continually in order to be used in prediction for the future. In our thesis, we have used a dataset of one of KAHRAMAA water stations to predict the water loss by adding some new feature to enhance the prediction process. We have studied the prediction of water loss for one-step and multi-step ahead using various classification, regression and ensemble machine learning strategies, algorithms and features.

We have experimented with different machine learning algorithms applied to multivariate input features, we have studied the response of time windowing on accuracy and root mean squared error. Based on the results of the experiments we have conclude the following:

- The proposed combination between bagging ensemble method and random forest model resulted in better results in most results compared to all other models.
- The highest accuracy for predicting water loss in next hour is 84.78% using combination between bagging ensemble method and J48 model with RMSE 0.37.

The result surpass the baseline models by 14.78% in accuracy.

- The highest accuracy for predicting water loss in 12th hour is 73.01% using combination between bagging ensemble method and random forest model with RMSE 0.5. The result surpass the baseline models by 45.32% in accuracy.
- The highest accuracy for predicting water loss in 24th hour is 71.66% using combination between bagging ensemble method and random forest model with RMSE 0.37. The result surpass the baseline models by 11.50% in accuracy.
- Ensemble classification models outperforms the regression models for predicting the RPS water loss across all horizons.
- RMSE has the same pattern across horizons, it starts minimal at horizon 1, goes up in horizon 12 and finally goes down in horizon 24.

Future Work

The proposed forecasting models can be integrated with KM SCADA system to it will ease the manual calculations used by the SCADA engineer. The current forecasting models could be enhanced in several ways: (1) having data from other water stations, (2) use the models for systems outside the RPS stations, (3) Increase the dataset size and (4) use the model in different fluid systems.

(1) Having data from other water stations: In our study we focused in one RPS water station which is the Airport station, however different water stations could be integrated to have a study that may affect the forecasting results and accuracy.

(2) Use the models for systems outside the RPS stations: the RPS water stations are

mostly connected via pipes and valves, this also can be integrated using the proposed model to enhance the forecasting results.

(3) Increase the dataset size: by adding more yearly time based data into the model, the forecasting results can be enhanced and more accurate.

(4) Use the model in different fluid systems: Different fluid systems can apply the model such as oil and sewage stations since the drinking water normally has the same connections and systems. This could give different analysis and results.

REFERENCES

Qatar's Sustainable Development. (2012). *Qatar Energy Industry Sector Sustainability Report 2012*. Qatar, Doha. Retrieved from http://www.hse-reg-dg.com/qeisr2012/WWW/assets/downloads/Qatar_2012.pdf, on May 6, 2014.

General Secretariat Development Planning. (2011). *Qatar National Development Strategy 2011~2016*. Qatar, Doha. Retrieved from http://www.gsdp.gov.qa/gsdp_vision/docs/NDS_EN.pdf

Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). *Introduction to time series analysis and forecasting*. John Wiley & Sons.

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Candelieri, A., Conti, D., & Archetti, F. (2014). A graph based analysis of leak localization in urban water networks. *Procedia Engineering*, *70*, 228-237.

Lijuan, W., Hongwei, Z., & Hui, J. (2012). A leak detection method based on EPANET and genetic algorithm in water distribution systems. In *Software Engineering and Knowledge Engineering: Theory and Practice* (pp. 459-465). Springer Berlin Heidelberg.

Hou, C. X., & Zhang, E. H. (2013). Pipeline Leak Detection Based on Double Sensor Negative Pressure Wave. In *Applied Mechanics and Materials* (Vol. 313, pp. 1225-1228). Trans Tech Publications.

Chen, H., Ye, H., Chen, L. V., & Su, H. (2004, May). Application of support vector machine learning to leak detection and location in pipelines. In *Instrumentation and Measurement Technology Conference, 2004. IMTC 04. Proceedings of the 21st IEEE* (Vol. 3, pp. 2273-2277). IEEE.

Zhao, J., Li, D., Qi, H., Sun, F., & An, R. (2010, August). The fault diagnosis method of pipeline leakage based on neural network. In *2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering* (Vol. 1, pp. 322-325). IEEE.

Salam, A. (2015). Application Extreme Learning Machine to Predict Location and Magnitude of Pipe Leak on Water Distribution Network.

Nasir, M. T., Mysorewala, M., Cheded, L., Siddiqui, B., & Sabih, M. (2014, February). Measurement error sensitivity analysis for detecting and locating leak in pipeline using ANN and SVM. In *Systems, Signals & Devices (SSD), 2014 11th International Multi-Conference on* (pp. 1-4). IEEE.

Candelieri, A., Soldi, D., Conti, D., & Archetti, F. (2014). Analytical leakages localization in water distribution networks through spectral clustering and support vector machines. The icewater approach. *Procedia Engineering*, *89*, 1080-1088.

Herrera, M., Torgo, L., Izquierdo, J., & Pérez-García, R. (2010). Predictive models for forecasting hourly urban water demand. *Journal of hydrology*, *387*(1), 141-150.

Kim, J. H., Hwang, S. H., & Shin, H. S. (2001). A neuro-genetic approach for daily water demand forecasting. *KSCE Journal of Civil Engineering*, *5*(3), 281-288.

Bakker, M., Van Duist, H., Van Schagen, K., Vreeburg, J., & Rietveld, L. (2014). Improving the performance of water demand forecasting models by using weather input. *Procedia Engineering*, *70*, 93-102.

Benhmed, K., Shaban, K. B., & El-Hag, A. (2014, May). Cost effective assessment of transformers using machine learning approach. In *2014 IEEE Innovative Smart Grid Technologies-Asia (ISGT ASIA)* (pp. 328-332). IEEE.

Tan, P. N. (2006). *Introduction to data mining*. Pearson Education India.

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Hall, M., Witten, I., & Frank, E. (2011). Data mining: Practical machine learning tools and techniques. *Kaufmann, Burlington*.

Shevade, S. K., Keerthi, S. S., Bhattacharyya, C., & Murthy, K. R. K. (2000). Improvements to the SMO algorithm for SVM regression. *IEEE transactions on neural networks*, *11*(5), 1188-1193.

Quinlan, J. R. (1993). C4. 5: Programming for machine learning. *Morgan Kauffmann*, 38.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Freund, Y., & Schapire, R. E. (1996, July). Experiments with a new boosting algorithm. In *Icml* (Vol. 96, pp. 148-156).

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140.

APPENDIX

Table 6

PTA Percentage and RMSE using Regression Models.

Window Size	**Horizon**	**Model**	**Root Mean Squared Error**	**PTA (%)**
Not windowed	-	Linear Regression	218.200	67.5
		Multilayer Perceptron	230.775	63.6
		SMOReg	-	-
Window 24 Step 12	1	Linear Regression	242.243	86.2
	12		193.605	84.6
	24		463.29	76.9
	1	Multilayer Perceptron	974.767	46.2
	12		900.426	60
	24		1008.563	58.5
	1	SMOReg	139.317	89.2
	12		186.896	84.6
	24		181.195	81.5

Table 7

Accuracy & RMSE for Classification Models by Discretizing the Data into 3 Bins.

Window Size	Horizon	Model	Accuracy	RMSE	Confusion Matrix
Window 24 Step 12	1	J48	76.16% +/- 4.19%	0.477 +/- 0.040	a: a [-∞ - 426.771] 190, 20, 4; b [426.771 - 559.916] 25, 145, 47; c [559.916 - ∞] 6, 56, 170
		Random Forest	76.91% +/- 3.65%	0.431 +/- 0.025	a: a [-∞ - 426.771] 188, 23, 8; b [426.771 - 559.916] 29, 151, 42; c [559.916 - ∞] 4, 47, 171
		Random Tree	65.01% +/- 5.56%	0.590 +/- 0.046	a: a [-∞ - 426.771] 161, 43, 9; b [426.771 - 559.916] 41, 114, 56; c [559.916 - ∞] 19, 64, 156
	12	J48	55.75% +/- 5.68%	0.640 +/- 0.036	a: a [-∞ - 410.455] 144, 48, 27; b [410.455 - 531.257] 51, 98, 67; c [531.257 - ∞] 25, 75, 127
		Random Forest	62.10% +/- 5.96%	0.54 +/- 0.029	a: a [-∞ - 410.455] 156, 50, 19; b [410.455 - 531.257] 47, 117, 64; c [531.257 - ∞] 17, 54, 138
		Random Tree	55.60% +/- 4.24%	0.666 +/- 0.032	a: a [-∞ - 410.455] 149, 50, 28; b [410.455 - 531.257] 45, 104, 78; c [531.257 - ∞] 26, 67, 115
	24	J48	56.58% +/- 5.32%	0.639 +/- 0.035	a: a [-∞ - 411.628] 135, 47, 34; b [411.628 - 531.257] 50, 105, 53; c [531.257 - ∞] 35, 68, 134
		Random Forest	62.03% +/- 4.77%	0.544 +/- 0.031	a: a [-∞ - 411.628] 148, 45, 22; b [411.628 - 531.257] 49, 123, 60; c [531.257 - ∞] 23, 52, 139
		Random Tree	55.07% +/- 6.30%	0.668 +/- 0.049	a: A[-∞ - 411.628] 138, 59, 40; b [411.628 - 531.257] 46, 104, 59; c [531.257 - ∞] 36, 57, 122

Table 8

Accuracy and RMSE for Classification Models by Discretizing the Data into 4 Bins.

Window Size	Horizon	Model	Accuracy	RMSE	Confusion Matrix				
Window 24 Step 12	1	J48	58.85% +/- 5.32%	0.624 +/- 0.041		a	b	c	d
					a [-∞ - 367.026]	121	33	8	2
					b [367.026 - 491.137]	35	75	45	12
					c [491.137 - 608.578]	8	44	81	39
					d [608.578 - ∞]	1	14	32	113
		Random Forest	60.95% +/- 6.89%	0.548 +/- 0.034		a	b	c	d
					a [-∞ - 367.026]	141	37	4	4
					b [367.026 - 491.137]	20	83	58	7
					c [491.137 - 608.578]	4	36	63	38
					d [608.578 - ∞]	0	10	41	117
		Random Tree	54.77% +/- 8.48%	0.669 +/- 0.067		a	b	c	d
					a [-∞ - 367.026]	107	33	6	3
					b [367.026 - 491.137]	49	69	42	18
					c [491.137 - 608.578]	8	55	80	38
					d [608.578 - ∞]	1	9	38	107
	12	J48	47.27% +/- 6.23%	0.708 +/- 0.038		a	b	c	d
					a [-∞ - 367.026]	107	29	13	14
					b [367.026 - 491.137]	32	63	53	18
					c [491.137 - 608.578]	16	53	55	46
					d [608.578 - ∞]	10	21	44	88
		Random Forest	53.78% +/- 3.31%	0.609 +/- 0.016		a	b	c	d
					a [-∞ - 367.026]	116	26	15	9
					b [367.026 - 491.137]	30	80	51	20
					c [491.137 - 608.578]	11	40	61	38
					d [608.578 - ∞]	8	20	38	99
		Random Tree	46.53% +/- 6.10%	0.73 +/- 0.042		a	b	c	d
					a [-∞ - 367.026]	100	24	11	14
					b [367.026 - 491.137]	28	68	52	19
					c [491.137 - 608.578]	22	51	56	49
					d [608.578 - ∞]	15	23	46	84
	24	J48	44.34% +/- 4.57%	0.729 +/- 0.028		a	b	c	d
					a [-∞ - 367.026]	105	31	17	11
					b [367.026 - 491.137]	27	50	54	24
					c [491.137 - 608.578]	20	64	54	47
					d [608.578 - ∞]	13	20	40	84
		Random Forest	50.38% +/- 6.33%	0.636 +/- 0.027		a	b	c	d
					a [-∞ - 367.026]	113	35	16	11
					b [367.026 - 491.137]	31	68	56	15
					c [491.137 - 608.578]	12	47	56	44
					d [608.578 - ∞]	9	15	37	96
		Random Tree	43.43% +/- 6.59%	0.751 +/- 0.045		a	b	c	d
					a [-∞ - 367.026]	90	34	18	13
					b [367.026 - 491.137]	38	61	58	29
					c [491.137 - 608.578]	21	47	52	40
					d [608.578 - ∞]	16	23	37	84

Table 9

Accuracy and RMSE for Boosting Ensemble Models by Discretizing the Data into 3 Bins

Window Size	Horizon	Model	Accuracy	RMSE	Confusion Matrix
Window 24 Step 12	1	J48	78.58% +/- 3.64%	0.445 +/- 0.040	a b c a [-∞ - 426.771] 185 18 4 b [426.771 - 559.916] 30 165 46 c [559.916 - ∞] 6 38 171
		Random Forest	79.34% +/- 2.35%	0.449 +/- 0.025	a b c a [-∞ - 426.771] 186 20 3 b [426.771 - 559.916] 34 163 41 c [559.916 - ∞] 1 38 177
		Random Tree	66.02% +/- 8.29%	0.579 +/- 0.071	a b c a [-∞ - 426.771] 158 31 14 b [426.771 - 559.916] 45 128 55 c [559.916 - ∞] 18 62 152
	12	J48	66.62% +/- 5.29%	0.562 +/- 0.042	a b c a [-∞ - 410.455] 154 28 13 b [410.455 - 531.257] 48 131 52 c [531.257 - ∞] 18 62 156
		Random Forest	64.37% +/- 6.57%	0.583 +/- 0.052	a b c a [-∞ - 410.455] 150 40 22 b [410.455 - 531.257] 56 135 58 c [531.257 - ∞] 14 46 141
		Random Tree	51.97% +/- 6.11%	0.692 +/- 0.044	a b c a [-∞ - 410.455] 132 50 27 b [410.455 - 531.257] 57 96 78 c [531.257 - ∞] 31 75 116
	24	J48	64.14% +/- 3.50%	0.588 +/- 0.036	a b c a [-∞ - 411.628] 150 38 18 b [411.628 - 531.257] 45 124 53 c [531.257 - ∞] 25 58 150
		Random Forest	62.78% +/- 5.12%	0.597 +/- 0.043	a b c a [-∞ - 411.628] 154 55 19 b [411.628 - 531.257] 47 123 64 c [531.257 - ∞] 19 42 138
		Random Tree	54.00% +/- 6.91%	0.676 +/- 0.050	a b c a [-∞ - 411.628] 142 59 41 b [411.628 - 531.257] 45 93 58 c [531.257 - ∞] 33 68 122

Table 10

Accuracy and RMSE for Boosting Ensemble Models by Discretizing the Data into 4 Bins.

Window Size	Horizon	Model	Accuracy	RMSE	Confusion Matrix
Window 24 Step 12	1	J48	64.43% +/- 7.22%	0.579 +/- 0.061	a: a [-∞ - 367.026] 125, b 25, c 0, d 2 b: b [367.026 - 491.137] 35, 90, 43, 7 c: c [491.137 - 608.578] 4, 46, 91, 36 d: d [608.578 - ∞] 1, 5, 32, 121
		Random Forest	62.76% +/- 5.82%	0.601 +/- 0.046	a: a [-∞ - 367.026] 134, b 30, c 3, d 3 b: b [367.026 - 491.137] 28, 91, 54, 9 c: c [491.137 - 608.578] 3, 40, 72, 35 d: d [608.578 - ∞] 0, 5, 37, 119
		Random Tree	53.28% +/- 8.92%	0.680 +/- 0.064	a: a [-∞ - 367.026] 111, b 32, c 4, d 2 b: b [367.026 - 491.137] 33, 67, 47, 14 c: c [491.137 - 608.578] 11, 50, 74, 49 d: d [608.578 - ∞] 10, 17, 41, 101
	12	J48	52.43% +/- 5.90%	0.671 +/- 0.041	a: a [-∞ - 361.468] 114, b 21, c 9, d 8 b: b [361.468 - 473.306] 30, 78, 57, 16 c: c [473.306 - 577.658] 14, 54, 55, 42 d: d [577.658 - ∞] 7, 13, 44, 100
		Random Forest	55.00% +/- 3.40%	0.661 +/- 0.026	a: a [-∞ - 361.468] 117, b 24, c 6, d 11 b: b [361.468 - 473.306] 32, 84, 59, 14 c: c [473.306 - 577.658] 9, 47, 60, 38 d: d [577.658 - ∞] 7, 11, 40, 103
		Random Tree	47.74% +/- 5.06%	0.722 +/- 0.034	a: a [-∞ - 361.468] 99, b 32, c 15, d 11 b: b [361.468 - 473.306] 28, 66, 41, 17 c: c [473.306 - 577.658] 22, 43, 62, 49 d: d [577.658 - ∞] 16, 25, 47, 89
	24	J48	49.63% +/- 6.14%	0.683 +/- 0.047	a: a [-∞ - 362.376] 109, b 26, c 11, d 5 b: b [362.376 - 473.306] 33, 57, 47, 23 c: c [473.306 - 577.658] 14, 60, 64, 40 d: d [577.658 - ∞] 9, 22, 43, 98
		Random Forest	51.44% +/- 3.59%	0.686 +/- 0.026	a: a [-∞ - 362.376] 118, b 31, c 11, d 9 b: b [362.376 - 473.306] 26, 69, 60, 17 c: c [473.306 - 577.658] 11, 55, 53, 40 d: d [577.658 - ∞] 10, 10, 41, 100
		Random Tree	45.99% +/- 4.47%	0.734 +/- 0.031	a: a [-∞ - 362.376] 96, b 29, c 26, d 12 b: b [362.376 - 473.306] 33, 57, 46, 20 c: c [473.306 - 577.658] 21, 58, 59, 42 d: d [577.658 - ∞] 15, 21, 34, 92

Table 11

Accuracy and RMSE for Bagging Ensemble Models by Discretizing the Data into 3 Bins.

Window Size	Horizon	Model	Accuracy	RMSE	Confusion Matrix			
Window 24 Step 12	1	J48	81.14% +/- 3.64%	0.403 +/- 0.033		a	b	c
					a [-∞ - 426.771]	190	18	5
					b [426.771 - 559.916]	27	164	32
					c [559.916 - ∞]	4	39	184
		Random Forest	79.19% +/- 3.48%	0.429 +/- 0.018		a	b	c
					a [-∞ - 426.771]	187	23	2
					b [426.771 - 559.916]	29	151	32
					c [559.916 - ∞]	5	47	187
		Random Tree	76.91% +/- 3.21%	0.431 +/- 0.023		a	b	c
					a [-∞ - 426.771]	188	23	8
					b [426.771 - 559.916]	29	151	42
					c [559.916 - ∞]	4	47	171
	12	J48	64.67% +/- 5.67%	0.521 +/- 0.027		a	b	c
					a [-∞ - 410.455]	153	38	12
					b [410.455 - 531.257]	46	124	58
					c [531.257 - ∞]	21	59	151
		Random Forest	67.99% +/- 5.02%	0.521 +/- 0.021		a	b	c
					a [-∞ - 410.455]	159	35	13
					b [410.455 - 531.257]	47	131	48
					c [531.257 - ∞]	14	55	160
		Random Tree	62.10% +/- 3.26%	0.54 +/- 0.024		a	b	c
					a [-∞ - 410.455]	156	50	19
					b [410.455 - 531.257]	47	117	64
					c [531.257 - ∞]	17	54	138
	24	J48	63.24% +/- 4.59%	0.541 +/- 0.024		a	b	c
					a [-∞ - 458.681]	146	38	15
					b [458.681 - 599.894]	52	125	59
					c [599.894 - ∞]	22	57	147
		Random Forest	65.96% +/- 5.70%	0.533 +/- 0.021		a	b	c
					a [-∞ - 411.628]	151	35	13
					b [411.628 - 531.257]	50	133	56
					c [531.257 - ∞]	19	52	152
		Random Tree	62.03% +/- 5.41%	0.544 +/- 0.032		a	b	c
					a [-∞ - 411.628]	148	45	22
					b [411.628 - 531.257]	49	123	60
					c [531.257 - ∞]	23	52	139

Table 12

Accuracy and RMSE for Bagging Ensemble Models by Discretizing the Data into 4 Bins.

Window Size	Horizon	Model	Accuracy	RMSE	Confusion Matrix				
Window 24 Step 12	1	J48	65.50% +/- 6.75%	0.510 +/- 0.031		a	b	c	d
					a [-∞ - 367.026]	133	23	4	3
					b [367.026 - 491.137]	27	100	48	8
					c [491.137 - 608.578]	4	40	81	35
					d [608.578 - ∞]	1	3	33	120
		Random Forest	63.37% +/- 6.47%	0.542 +/- 0.034		a	b	c	d
					a [-∞ - 367.026]	140	28	2	3
					b [367.026 - 491.137]	20	81	48	6
					c [491.137 - 608.578]	5	50	73	31
					d [608.578 - ∞]	0	7	43	126
		Random Tree	60.95% +/- 5.14%	0.548 +/- 0.035		a	b	c	d
					a [-∞ - 367.026]	141	37	4	4
					b [367.026 - 491.137]	20	83	58	7
					c [491.137 - 608.578]	4	36	63	38
					d [608.578 - ∞]	0	10	41	117
	12	J48	53.62% +/- 3.88%	0.611 +/- 0.013		a	b	c	d
					a [-∞ - 361.468]	116	24	7	9
					b [361.468 - 473.306]	29	74	52	12
					c [473.306 - 577.658]	10	53	61	41
					d [577.658 - ∞]	10	15	45	104
		Random Forest	56.06% +/- 3.67%	0.608 +/- 0.010		a	b	c	d
					a [-∞ - 361.468]	121	25	4	6
					b [361.468 - 473.306]	25	82	59	12
					c [473.306 - 577.658]	10	46	52	32
					d [577.658 - ∞]	9	13	50	116
		Random Tree	53.78% +/- 2.79%	0.609 +/- 0.016		a	b	c	d
					a [-∞ - 361.468]	116	26	15	9
					b [361.468 - 473.306]	30	80	51	20
					c [473.306 - 577.658]	11	40	61	38
					d [577.658 - ∞]	8	20	38	99
	24	J48	49.93% +/- 4.57%	0.629 +/- 0.022		a	b	c	d
					a [-∞ - 362.376]	114	25	11	9
					b [362.376 - 473.306]	30	63	57	14
					c [473.306 - 577.658]	9	53	46	36
					d [577.658 - ∞]	12	24	51	107
		Random Forest	54.02% +/- 5.58%	0.622 +/- 0.020		a	b	c	d
					a [-∞ - 362.376]	114	30	8	7
					b [362.376 - 473.306]	34	71	53	17
					c [473.306 - 577.658]	12	49	64	34
					d [577.658 - ∞]	5	15	40	108
		Random Tree	50.38% +/- 6.11%	0.636 +/- 0.025		a	b	c	d
					a [-∞ - 362.376]	113	35	16	11
					b [362.376 - 473.306]	31	68	56	15
					c [473.306 - 577.658]	12	47	56	44
					d [577.658 - ∞]	9	15	37	96

Table 13

Validation of Classification Models vs Baseline for Discretizing Data into 3 Bin and Window24, Step 12, and Horizon 1.

	baseline	RF	J48	RT	Bagging (RF)	Bagging (J48)	Bagging (RT)	Boosting (RF)	Boosting (J48)	Boosting (RT)
Correctly classified as "a"	156	188	190	161	187	190	188	186	185	158
Correctly classified as "b"	124	151	145	114	151	164	151	163	165	128
Correctly classified as "c"	160	171	170	156	187	184	171	177	171	152
Total Correctly Classified Instances	440	510	505	431	525	538	510	526	521	438
Correctly classified in percentage (%)	66.365	76.923	76.168	65.007	79.185	81.146	76.923	79.336	78.582	66.063
Difference between algorithm & baseline (%)		10.558	9.803	-1.357	12.820	14.781	10.558	12.971	12.217	-0.301

Table 14

Validation of Classification Models vs Baseline for Discretizing Data into 3 Bins and Window24, Step 12, and Horizon 12.

	baseline	RF	J48	RT	Bagging (RF)	Bagging (J48)	Bagging (RT)	Boosting (RF)	Boosting (J48)	Boosting (RT)
Correctly classified as "a"	42	156	144	149	159	153	156	150	154	132
Correctly classified as "b"	64	117	98	104	131	124	117	135	131	96
Correctly classified as "c"	44	138	127	115	160	151	138	141	156	116
Total Correctly Classified Instances	150	411	369	368	450	428	411	426	441	344
Correctly classified in percentage (%)	22.658	62.084	55.740	55.589	67.975	64.652	62.084	64.350	66.616	51.963
Difference between algorithm & baseline (%)		39.425	33.081	32.930	45.317	41.993	39.425	41.691	43.957	29.305

Table 15

Validation of Classification Models vs Baseline for Discretizing Data into 3 Bin and Window24, Step 12, and Horizon 24.

	baseline	RF	J48	RT	Bagging (RF)	Bagging (J48)	Bagging (RT)	Boosting (RF)	Boosting (J48)	Boosting (RT)
Correctly classified as "a"	136	148	135	138	151	146	148	154	150	142
Correctly classified as "b"	106	123	105	104	133	125	123	123	124	93
Correctly classified as "c"	118	139	134	122	152	147	139	138	150	122
Total Correctly Classified Instances	360	410	374	364	436	418	410	415	424	357
Correctly classified in percentage (%)	54.462	62.027	56.580	55.068	65.960	63.237	62.027	62.783	64.145	54.009
Difference between algorithm & baseline (%)		7.564	2.118	0.605	11.497	8.774	7.564	8.320	9.682	-0.453

Table 16

Validation of Classification Models vs. Baseline for Discretizing Data into 4 Bins and Window24, Step 12, and Horizon 1.

	baseline	RF	J48	RT	Bagging (RF)	Bagging (J48)	Bagging (RT)	Boosting (RF)	Boosting (J48)	Boosting (RT)
Correctly classified as "a"	108	141	121	107	140	133	141	134	125	111
Correctly classified as "b"	68	83	75	69	81	100	83	91	90	67
Correctly classified as "c"	66	63	81	80	73	81	63	72	91	74
correctly classified as "d"	105	117	113	107	126	120	117	119	121	101
Total Correctly Classified Instances	347	404	390	363	420	434	404	416	427	353
Correctly classified in percentage (%)	52.337	60.935	58.823	54.751	63.348	65.460	60.935	62.745	64.404	53.242
Difference between algorithm & baseline (%)		8.597	6.485	2.413	11.010	13.122	8.597	10.407	12.066	0.904

Table 17

Validation of Classification Models vs Baseline for Discretizing Data into 4 Bins and Window24, Step 12, and Horizon 12.

	baseline	RF	J48	RT	Bagging (RF)	Bagging (J48)	Bagging (RT)	Boosting (RF)	Boosting (J48)	Boosting (RT)
Correctly classified as "a"	10	116	107	100	121	116	116	117	114	99
Correctly classified as "b"	49	80	63	68	82	74	80	84	78	66
Correctly classified as "c"	35	61	55	56	52	61	61	60	55	62
correctly classified as "d"	22	99	88	84	116	104	99	103	100	89
Total Correctly Classified Instances	116	356	313	308	371	355	356	364	347	316
Correctly classified in percentage (%)	17.522	53.776	47.280	46.525	56.042	53.625	53.776	54.984	52.416	47.734
Difference between algorithm & baseline (%)		36.253	29.758	29.003	38.519	36.102	36.253	37.462	34.894	30.211

Table 18

Validation of Classification Models vs Baseline for Discretizing Data into 4 Bins and Window24, Step 12, Horizon 24.

	baseline	RF	J48	RT	Bagging (RF)	Bagging (J48)	Bagging (RT)	Boosting (RF)	Boosting (J48)	Boosting (RT)
Correctly classified as "a"	90	113	105	90	114	114	113	118	109	96
Correctly classified as "b"	67	68	50	61	71	63	68	69	57	57
Correctly classified as "c"	54	56	54	52	64	46	56	53	64	59
correctly classified as "d"	74	96	84	84	108	107	96	100	98	92
Total Correctly Classified Instances	285	333	293	287	357	330	333	340	328	304
Correctly classified in percentage (%)	43.116	50.378	44.326	43.419	54.009	49.924	50.378	51.437	49.621	45.990
Difference between algorithm & baseline (%)		7.261	1.210	0.302	10.892	6.807	7.261	8.320	6.505	2.874