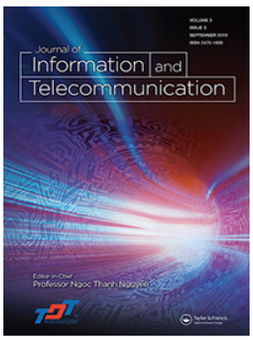

Taylor & Francis
Taylor & Francis Group

Journal of Information and Telecommunication

ISSN: 2475-1839 (Print) 2475-1847 (Online) Journal homepage: https://www.tandfonline.com/loi/tjit20

Machine learning approaches for anomaly detection of water quality on a real-world data set

Fitore Muharemi, Doina Logofătu & Florin Leon

To cite this article: Fitore Muharemi, Doina Logofătu & Florin Leon (2019) Machine learning approaches for anomaly detection of water quality on a real-world data set, Journal of Information and Telecommunication, 3:3, 294-307, DOI: 10.1080/24751839.2019.1565653

To link to this article: https://doi.org/10.1080/24751839.2019.1565653

Published online: 03 Feb 2019.

JOURNAL OF INFORMATION AND TELECOMMUNICATION
2019, VOL. 3, NO. 3, 294–307
https://doi.org/10.1080/24751839.2019.1565653

Machine learning approaches for anomaly detection of water quality on a real-world data set*

Fitore Muharemi[a], Doina Logofătu[a] and Florin Leon[b]

[a]Faculty of Computer Science and Engineering, Frankfurt University of Applied Sciences, Frankfurt Am Main, Germany; [b]Computer Science and Engineering, Technical University of Iași, Iași, Romania

ABSTRACT

Accurate detection of water quality changes is a crucial task of water companies. Water supply companies must provide safe drinking water. Nowadays in different areas, we find sensible sensors which monitor data during the time. Normally the data registered by the sensors contain a meaning, such as there can be any event. Sometimes the data are ill-understood and stating if there is an event which is difficult. This work represents the description of several approaches to identifying changes or anomalies occurring on water quality time series data. This work also discusses and proposes a solution to some challenges when dealing with time series data. The following models are applied to water quality data: logistic regression, linear discriminant analysis, support vector machines (SVM), artificial neural network (ANN), deep neural network (DNN), recurrent neural network (RNN) and long short-term memory (LSTM). The performance evaluation is conducted using F-score metric. A simulation study is conducted to check the performance of each algorithm using F-score. Solving imbalanced data is basically intentionally biasing the data to get interesting results instead of accurate results. The results show that all algorithms are vulnerable although SVM, ANN and logistic regressions tend to be a little less vulnerable, while DNN, RNN and LSTM are very vulnerable.

ARTICLE HISTORY
Received 30 June 2018
Accepted 4 January 2019

KEYWORDS
Classification; water quality; F1 score; imputation; event

1. Introduction

Water covers 71% of Earth's surface and is vital for all known forms of life. The purity of drinking water is an essential task for water supply companies all over the world, and today this is a well-known problem because of many vulnerable attacks. The change of water chemistry can happen prior earthquakes, due to terrorist attacks, or other pollution caused by man-made. The change of the water chemistry by the earthquake is a good indicator of earthquake predictions, but in the other hand, these changes can be very harmful to human health as some of the changes have a toxic nature. Because of all these risks in public health, the anomaly detection of water systems has a high importance. To control the drinking water quality today, water companies have employed contamination warning systems. By using different sensors they monitor relevant water quality and environmental data at

*This study extends the work in Muharemi, Logofătu, Andersson, and Leon (2018a)

CONTACT Fitore Muharemi muharemi@stud.fra-uas.de

several measuring points on a regular basis. Despite these measurements, the system also needs a detection system which accurately notifies for water quality changes based on measured values. The Goal of the GECCO 2017 Industrial Challenge[1] was to develop a change detection system to accurately predict any kind of change in time series of drinking water composition data. An adequate and accurate alarm system that allows for early recognition of all kind of changes is a basic requirement for the provision of clean and safe drinking water.

A well-known problem when dealing with the real-world data is that the data are noisy and highly imbalanced. Highly imbalanced data sets have proven difficult to explain and predict. Researches on imbalanced classes often consider imbalanced to mean a minority class of 10–20%, but in reality, data sets can get far more imbalanced than this. For example, factory production defect rates typically run about 0.1%; about 2% of credit card accounts are defrauded per year; medical screening for a condition is usually performed on a large population of people without the condition, to detect a small minority with it (e.g. HIV prevalence in the USA is around 0.4%)[2]. The data set we are using in this work is highly imbalanced, and it makes the prediction more difficult. This paper extends the ACIIDS 2018 paper Muharemi et al. (2018a) with the aim of providing better results for water quality prediction. Based on the related works, we investigate the fact that how well popular algorithms perform on this specific highly imbalanced data and compare the performance between statistical and machine learning algorithms on how they perform and how well they can predict and detect changes in this time series data set. The experiment was conducted in two parts. First, for the classification we use the statistical algorithm logistic regression (Winner model on GECCO IC 2017), while for the missing values we have been using simplest methods, mean value, filling with zero, or ignoring data with missing parameters (Chandrasekaran, Freise, Stork, Rebolledo, & Bartz-Beielstein, 2017; Muharemi et al., 2018a). Second, we extend the experiment by using machine learning techniques, ANN, DNN, RNN, LSTM, and LDA to compare if they can outperform the logistic regression for this specific data set. The methodology of solving the problem includes seven classification algorithms on the same classification task. When sufficiently representative training data were used, most algorithms perform reasonably well, but in our experiment even we had a large data set, not every algorithm gave many promising results. The goal of this work is to find the most suitable algorithm for the problem under investigation. There is a number of dimensions we can look at to give a sense of what will be a reasonable algorithm: number of training data, number of features, the dependency of features, etc. We also consider that feature selection is a very important stage to decide on the performance of the algorithm.

How to decide on which algorithm is fitting best? This is the most important answer we try to answer nowadays. There are many methods to help us decide to go with a tested model or not. In this research, we use the F1 score, considered as one of the best performance metrics for classification algorithms, especially good for imbalanced data. This research can serve for many water supplier companies.

The rest of paper is organized as follows. Section 2 outlines related works on the analysis made on water quality and multivariate time series. Section 3 briefly describes the data set. Sections 4 and 5 explain preprocessing and feature selection on our data set, respectively. Sections 6–8, respectively, present the tools used, experimental evaluation and results. In the last section is included the conclusion and future work.

2. Related work

Water as a very important factor in life makes very crucial devising new methodologies for analysing water quality and forecast future water quality trends. Many researchers have analysed the water quality problem. Byer and Carlson (2005) are among the first to create and test an online monitoring of drinking water distribution systems. They added four credible threat drinking water contaminants (aldicarb, sodium arsenate, sodium cyanide, and sodium fluoroacetate) to a tap water and analysed at different concentrations to determine their detectability in a distribution system. Benchtop analysis and online monitoring equipment were used to measure pH, chlorine residual, turbidity, and total organic carbon values before and after the introduction of these contaminants. Results indicate that all four contaminants can be detected at relatively low concentrations. Three of the four contaminants were detected below a concentration that would cause significant health effects [8].

Zhang, Zhu, Yue, and Wong (2017) propose a novel anomaly detection algorithm for water quality data using dual time-moving windows, which can identify anomaly data from historical patterns in real-time. The algorithm is based on statistical models, auto-regressive linear combination model. They have tested the algorithm using 3-month water quality data of PH from a real water quality monitoring station in a river system. Experimental results show that their algorithms can significantly decrease the rate of false positive and has better anomaly detection performance than AD and ADAM algorithms.

Mohammadpour et al. (2015) have investigated the problem with water quality, by using three different algorithms, SVM, and two methods of artificial neural networks. The performance is compared using R^2, RMSE, MAE. On the results they achieved, the SVM algorithm is competitive with neural networks. This work pushed us to remodel SVM and ANN from Muharemi et al. (2018a) as it promises to give better results.

Also Kang, Gao, and Xie (n.d.) have developed a model for water quality prediction. The best result was achieved using Artificial Neural Network with Nonlinear Autoregressive. In 2009, Xiang & Jiang applied least squares support vector machine (LS-SVM) with particle swarm optimization methods to predict the water quality and defeat the weaknesses of customary backpropagation algorithms as being moderate to meet and simple to achieve the extreme minimum value. They discovered that through simulation testing, the model shows high proficiency in estimating the water quality of the Liuxi River Xiang and Jiang (2009). The Recurrent Neural Network as a dynamical system, whose next state and output depend on the present network state and input, recently is applied in large-scale vision speech problems (Gregor, Danihelka, Graves, Rezende, & Wierstra, 2015). RNNs and LSTMs are pretty good at extracting patterns in input feature space, where the input data spans overlong sequences. They can almost seamlessly model problems with multiple input variables, this brings a great benefit in time series forecasting, where classical linear methods can be difficult to adapt to multivariate or multiple input forecasting problems (Che, Purushotham, Cho, Sontag, & Liu, 2018).

As we have seen, machine learning can achieve good results for anomaly detection of water quality, and our work is inspired by these related works. Machine learning algorithms can significantly decrease the number of false predictions.

3. Data description

This experiment extracts data from the public water company Thüringer Fernwasserversorgung, located at the heart of Germany. The Thüringer Fernwasserversorgung company performed measurements at significant points throughout the whole water distribution system, in particular at the outflow of the waterworks and the in- and outflow of the water towers. For this purpose, a part of the water is bypassed through a sensor system where the most important water quality indicators are measured. The data that is supplied for this challenge has been measured at different stations near the outflow of a waterworks. The data set we use in this work is time series, and it is composed of 122,334 registered samples with 6 water quality indicators, 3 operational data attributes and the label which indicates if there is an event or not. Table 1 provides the water quality indicators measured for this data set. The chlorine dioxide, the pH value, the redox potential, the electric conductivity and the turbidity of the water provide an indication for any changes on the water (event), while the flow rate and the temperature are considered as operational data, changes in these values may indicate changes in the related quality values but are not considered as events themselves. The EVENT is the target variable, which one we want to predict with a high accuracy. The possible values of EVENT are true or false, so we are dealing with a classification problem. The data set is a multivariate time series denoting water quality data and operative data on a minute basis. Real drinking-water time series are provided for training, testing, and assessing event detection methods. Same as for training, the testing set has three months collecting data. The period of the collection is different for the training and testing, from February to May and February to August, respectively. The data we use on this work belong to rare events data, binary dependent variables with dozens to thousands of times fewer ones (events, cases of fraudulent use, or epidemiological infections) than zeros ('nonevents') King and Zeng (2001). According to King and Zeng (2001), these variables have proven difficult to explain and predict, a problem that seems to have at least two sources. The training data we use are highly imbalanced with less than 2% of true events, and in such a problem maximizing the accuracy is meaningless if we assume that the rare class examples are much more important to classify, which is true for the problem we try to solve. Figure 1 shows the distribution of training data based on events detected in drinking water quality.

Figure 2 shows the box plots representing the distribution of features across two event modes. We see that the ability for the distinction between event modes is very low almost on all features, except the Redox, to some extent. So based on this plot, we conclude that to achieve a good result we must combine all features, as by using them independently there is not a good possibility to classify between two classes.

Table 1. Description of the given time series data.

ColumnName	Description
Time	Time of measurement, given in following format: yyyy-mm-dd HH:MM:SS
Tp	The temperature of the water, given in C.
Cl	Amount of chlorine dioxide in the water, given in mg/L (MS1)
pH	PH value of the water
Redox	Redox potential, given in mV
Leit	Electric conductivity of the water, given in μS/cm
Trueb	Turbidity of the water, given in NT
Cl_2	Amount of chlorine dioxide in the water, given in mg/L (MS2)
Fm	Flow rate at water line 1, given in m^3/h
Fm_2	Flow rate at water line 2, given in m^3/h
EVENT	remarkable change, given in boolean.

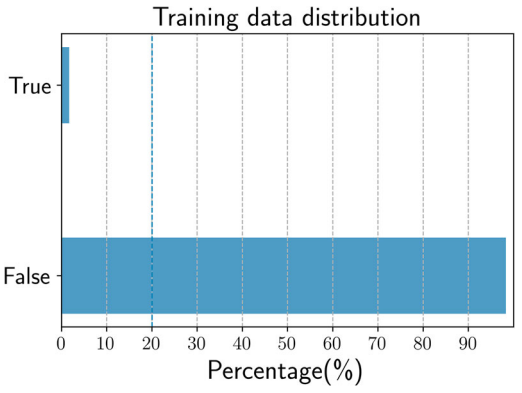

Figure 1. Distribution of the collected data based on event.

4. Data preparation

Unfortunately, real-world databases are highly influenced by negative factors such as the presence of noise, inconsistent and superfluous data and huge sizes in both dimensions, examples, and features. If data are not prepared, the results offered will not make sense, or often they are not as accurate as we expect. Some important steps in data preparation

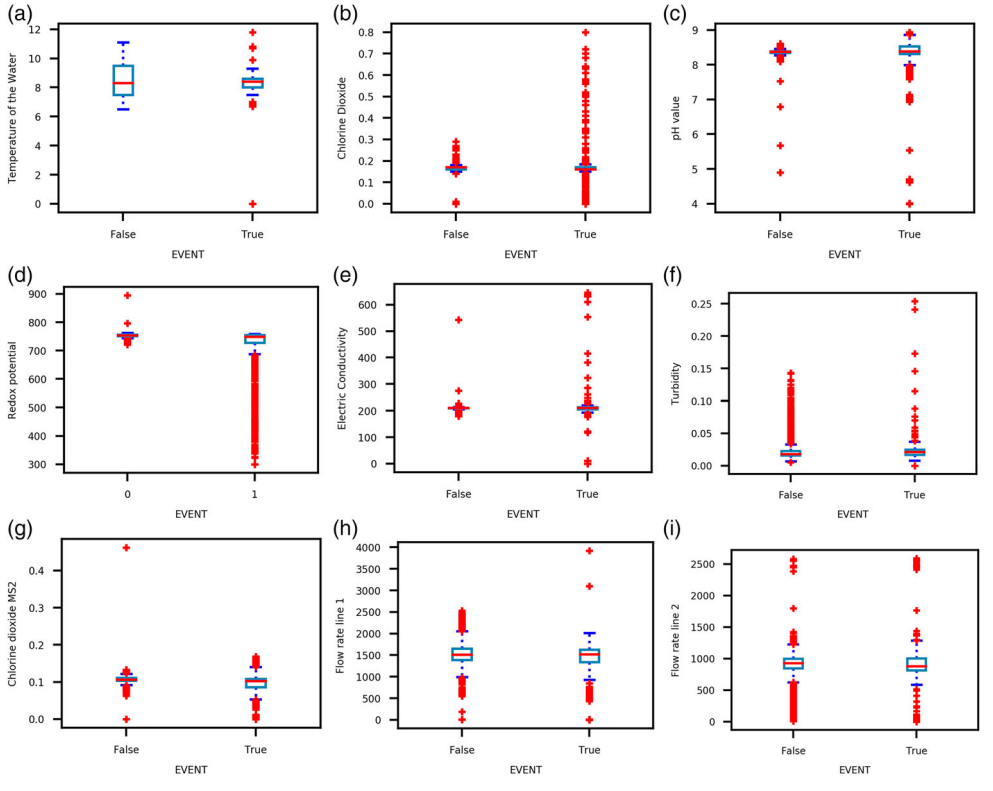

Figure 2. Distribution of water quality features over two event modes. (a) Temperature, (b) Chlorine dioxide, (c) pH value, (d) redox potential, (e) electric conductivity, (f) turbidity, (g) chlorine dioxide, (h) flow rate line 1, (i) flow rate line 2.

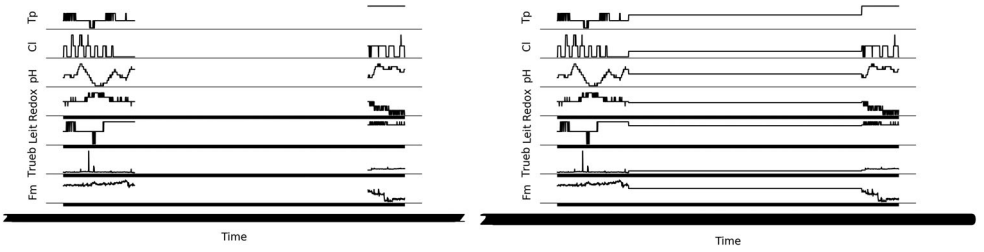

Figure 3. Date 30/08/2016 of time series, before and after imputation using mean of that day. (a) One day time series with missing values. (b) One day time series after imputation.

are: data cleaning, data transformation, data integration, data normalization, noise identification (García, Luengo, & Herrera, 2015). Missing values can be ignorable and non-ignorable, if we cannot ignore then we have to find an appropriate way to impute them. Our data set contains less than 1% missing values, but as our data set is a time series, normally is not suggested to ignore missing values, as the data are related to each other. In our experiment, we tried both approaches for handling the problem and based on the given values by the F1 score, the second approach proved to be more appropriate. There is a vast of methods of filling in the missing values, imputation of the mean, fill with zero, KNN-imputation, randomForest (Muharemi, Logofătu, & Leon, 2018b; Schafer, 2010), etc.

Based on the data analysis, we had missing values on three different days, and not a complete day was missing. So the imputation is done by using the mean of each time series on the same day, where there is no EVENT(false). In the first experiment, we filled missing values with zeros and trained using the logistic regression. In this case, the algorithm learned that when the complete sample is filled up with zero, the EVENT is false, so that is why filling with zero imputation method worked for our data set. To extend the experiment, we use a different imputation method which we expect to be more successful as it is a more intuitive approach. Better results are achieved using imputation with the mean value of false events from the same day, and we can see in Figure 3 on the left side is one-day time series with missing values, and on the right after imputation. We expect these values to be predicted correctly, as normally the events are happening when features are deviating from their normal range, and in this case, the model has to predict a False event.

5. Feature selection

We know that some machine learning algorithms can have poor performance if there are highly correlated data. From the Table 2, we see that Redox is positively correlated with pH, while Cl_2 is negatively correlated with Tp, and this can have a negative impact on the performance of the algorithms.

Increasing the number of features to modelling does not necessarily increase classifier accuracy since features may be redundant or not indicative of class (event/not event). Thus, feature selection is necessary to identify the essential features and eliminate redundant features. Feature selection methods can be categorized as filter methods, wrapper methods, or embedded methods Zhang and Sawchuk (2011). Filter methods use general characteristics of the data to evaluate features without involving a classifier in the process. A wrapper method is based on using accuracy from a specific classifier to

Table 2. Correlation table.

	Tp	Cl	pH	Redox	Leit	Trueb	Cl_2	Fm	Fm_2
Tp	1.0000	−0.0079	0.1941	0.0342	−0.2812	−0.3282	−0.4736	−0.3920	−0.0611
Cl	−0.0079	1.0000	0.1756	0.1222	0.0252	0.0181	0.1641	0.0255	0.0161
pH	0.1941	0.1756	1.0000	0.6501	−0.1544	−0.0701	0.1042	−0.0858	−0.0456
Redox	0.0342	0.1222	0.6501	1.0000	−0.1790	0.0179	0.4587	0.0349	0.0466
Leit	−0.2812	0.0252	−0.1544	−0.1790	1.0000	−0.0569	−0.0840	−0.0043	−0.0379
Trueb	−0.3282	0.0181	−0.070	0.0179	0.0569	1.0000	0.3247	0.3965	0.2004
Cl_2	−0.4736	0.1641	0.1042	0.4587	−0.0840	0.3247	1.0000	0.4093	0.1748
Fm	−0.3920	0.0255	−0.0858	0.0349	−0.0043	0.3965	0.4093	1.0000	0.2310
Fm_2	−0.0611	0.0161	−0.0456	0.0466	−0.0379	0.2004	0.1748	0.2310	1.0000

select features. Embedded methods incorporate feature selection as an internal mechanism of classifier's training process. Thus, both wrapper and embedded methods produce results that are specific to the classifier used for the task. Therefore, features weights, or feature subset selection, may only be useful to researchers using that particular classifier. Feature subsets with similar classifier performance to the full feature set should reduce computational burden, thus facilitating real-time implementations. Today we can find a vast number of algorithms on feature selection which can help on deciding on the importance of our predictor variables. For the feature selection task, we employ embedded methods which consist of using classifiers feature importance internal mechanism to measure information gain (the predictive power) of each feature and then select those with highest predictive power. Random Forest classifiers provide straightforward methods for feature selection: mean decrease impurity and mean decrease accuracy. The mean decrease impurity or sometimes called gini importance is defined as the total decrease in node impurity which is weighted by the probability of reaching that node which is approximated by the proportion of samples reaching that node averaged over all trees of the ensemble. Individual decision trees intrinsically perform feature selection by selecting appropriate split points. This information can be used to measure the importance of each feature; the basic idea is that the more often a feature is used in the split points of a tree, the more important that feature is. Features used at the top of the tree contribute to the final prediction decision of a larger fraction of the input samples. The expected fraction of the samples they contribute to can thus be used as an estimate of the relative importance of the features. This notion of importance can be extended to decision tree ensembles, in our case Random Forest, by simply averaging the feature importance of each tree.

An all relevant feature selection wrapper algorithm is also used on this data set. The Boruta algorithm is a wrapper built around the random forest classification algorithm. It tries to capture all the important, interesting features you might have in your dataset with respect to an outcome variable Figure 4.

First, it duplicates the dataset, and shuffle the values in each column (shadow features), after that it trains a Random Forest classifier. It uses also the Mean Decrease Accuracy or Mean Decrease Impurity- for deciding the importance of the features of the data set, the higher the score, the better or more important. While fitting a random forest model on the data set, it can get rid of attributes in each iteration, which does not perform well in the process. On the other hand, Boruta can find all features which are either strongly or weakly relevant to the decision variable. Figure 5 is the result of the feature importance using Boruta algorithm.

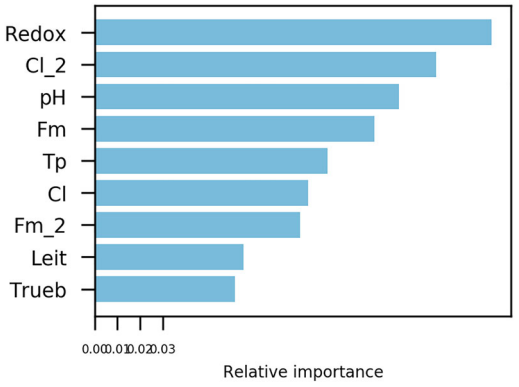

Figure 4. Feature importance using random forest.

Both algorithms for the feature importance we used, the embedded and wrapper method showed similar results.

6. Modelling tools

We have trained a set of models using classification algorithms: Logistic Regression, Support Vector Machines, Linear Discriminant Analysis, Neural Networks algorithm, Recurrent Neural Network, Deep Neural Network, and Long Short-Term Memory.

Logistic Regression is a popular machine learning algorithm. It is a simple algorithm that performs very well on a wide range of problems. Logistic regression is the appropriate regression analysis to conduct when the dependent variable is binary, as in our data set. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more independent variables. In our modelling, we tested this algorithm as it corresponds with the data we want to predict, predict TRUE or FALSE events. We have considered three different models using logistic regression. In two models, we have used attributes chosen as important from the two feature selection algorithms, and one model where we have used interaction terms (Altman, Marco, & Varetto, 1994; James, Witten, Hastie, & Tibshirani, 2013).

Linear Discriminant Analysis can be used to find a linear combination of features characterized by two or more events. LDA works well when measurements of the independent variables are continuous quantities. Normally classical pattern recognition techniques are

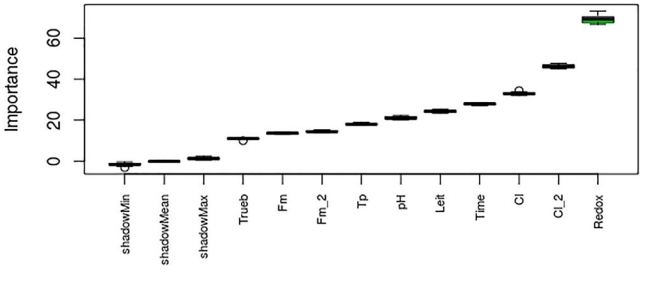

Figure 5. Feature importance using Boruta algorithm. Boxplot of each feature shows that the mean on each feature is higher than shadowMax (maximum *Z* score of a shadow attribute). None of the attributes is rejected.

a problem of great practical interest to experimental time series data. Time series classification problems are not restricted to geophysical applications but occur under many and varied circumstances in other fields. In our problem, it is not preferred much because of the assumptions about data, it assumes that data distribution is Gaussian (James et al., 2013). We tested its potential to our data set, and compared it with other techniques.

Artificial Neural Network (ANN) is known to be a good methodology for classification of complex data sets. The structure of this algorithm tends to simulate the structure of the human brain. The basic structure of an artificial neural network involves a network of many interconnected neurons. These neurons are very simple processing elements that individually combine pieces of one big problem. Each neuron computes one output using an activation function that considers the weighted sum of all its inputs. The most common activation function is the logistic sigmoid function: $f(x) = 1/1 + e^{-x}$, where $f(x)$ is the output of a neuron and x represents the weighted sum of inputs to a neuron. There are many different types of neural networks, but most well known are the multi-layer perceptron, Recurrent network(long short-term memory), Deep Neural Network (Yang, 2009). Figure 6 shows the plot of the simplest neural network model we tested.

Support Vector Machine (SVM) is a supervised machine learning algorithm introduced by Boser et al. (Boser, Guyon, & Vapnik, 1992), which can be used for both classification and regression problems. SVM is among the best 'off-the-shelf' supervised algorithm. It is a linear two-class classifier. The idea of the linear classifier is to find a hyperplane that can classify data points appropriately. SVM has the ability to accurately forecast time series data when the underlying system processes are typically nonlinear, non-stationary and not defined apriori. SVMs have also been proven to outperform other non-linear techniques including neural-network based non-linear prediction techniques such as multilayer perceptrons (Sapankevych & Sankar, 2009). It is also very good for binary classification, so it was expected to be good for our experiment. We tested different variants using SVM, by changing the parameters (kernel, cost, gamma). After parameters tuning, kernel='rbf', cost=500 and gamma=1 achieved one of the best F1 score.

Deep Neural Network (DNN) is a feedforward neural network with many hidden layers (Vincent, Larochelle, Bengio, & Manzagol, 2008). Deep learning methods aim at learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features. These models are called feedforward because information ows through the function being evaluated from x, through the intermediate computations used to dene f, and nally to the output y. There are no feedback connections in which outputs of the model is feed back into itself.

For our Deep Neural Network, we used three hidden layers of 6 neurons each and was trained for 200 epochs with a batch size of 15, with Adam optimizer and Sigmoid as the activation function. This model proved quite effective and stable.

Recurrent Neural Network (RNN) Unlike the feed-forward DNN, the RNN contains recurrent loops where the cells output state is feed back into input state. We decided to use RNN for our experiment as our intuition is that the nature of RNN can perfectly use the past information of time series. We choose to use a 75 unit hidden size, 10 layers deep RNN, with a single linear output layer. The RNN used Exponential Linear Units (ELU) activations and trained with a learning rate of 0.0001 and batch size of 200 at 2500 steps (Shipmon, Gurevitch, Piselli, & Edwards, 2017).

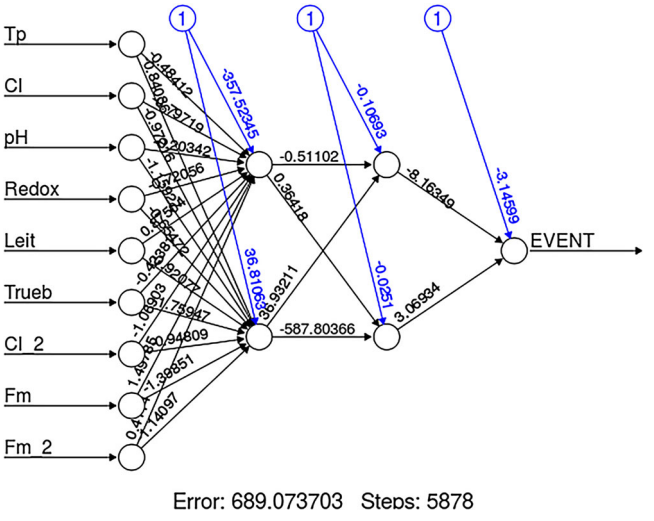

Figure 6. Neural Network plot from a sample of our data set. Input layer – nine predictor variables(left side). Two hidden layer, each with two neurons (middle). Output Layer (Response Variable).

Long Short-Term Memory (LSTM) is a form of RNN with a more complex cell architecture for more accurately maintaining the memory of important correlations. It was designed to model temporal sequences and their long-range dependencies more accurately than conventional RNNs. LSTM usually outperforms DNN (Sak, Senior, & Beaufays, 2014).

7. Evaluation

7.1. Metrics

Many real-life situations create highly imbalanced data, so nowadays imbalanced data learning is one of the challenging problems in data mining. The first intuition when checking the performance of a classifier is to look at the accuracy of that model as the number of correct predictions from all predictions made. A predictive model may have high accuracy, but be useless. In our data set, we have only 1.2% of the data. A trivial classifier that classifies all examples as non-events will achieve an accuracy of 98.8%, though its error rate for the minority class is 100%. The cost of diagnosing a true event as false can threat peoples life. When the performance on the minority class is as important or more important than overall accuracy, other performance measures must be used. For imbalanced data sets when the minority class is an important class, performance metrics mentioned by Bekkar, Djemaa, and Alitouche (2013) are often used. They are based on a confusion matrix that reports the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

Precision, Recall and *F*-score are used to evaluate the performance of a learner on the minority class (Bottenberg & Ward, 1963; Sokolova & Lapalme, 2009).

Precision: the number of correctly classified positive examples divided by the number of examples labelled by the system as positive

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Table 3. Algorithms performance using time series cross-validator.

Algorithm	True Pos	False Pos	True Neg	False Neg	F1
SVM	1696	0	137840	30	**0.9891**
DNN	1590	78	137346	136	0.9485
LSTM	1165	495	137345	555	0.9023
RNN	1096	800	136040	630	0.8345
LogRegression	924	416	137424	802	0.6027
Simple NN	842	4392	96914	884	0.5786
LDA	602	34673	103167	1124	0.0820

The best F1 score written in bold is achieved using the SVM model.

Recall: the number of correctly classified positive examples divided by the number of positive examples in the data

$$Recall = TPR = \frac{TP}{TP + FN} \tag{2}$$

F1 score: The harmonic mean of precision and recall.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

In this work, we focused on improving the F1 score, since it is the best metric to evaluate our experiment and also this metric is used for the selection of the best model by GECCO Industrial Challenge. Table 3 visualizes the F1 score of each trained model.

7.2. Model evaluation

Traditional *k*-fold cross-validation methods of model evaluation have limitations for time series data, as time series data is characterized by the correlation between observations that are near in time. To validate the classifiers, we use time series cross-validator, as it is very important to evaluate our model for time series data on 'future' observations least like those that are used to train the model. Time series cross-validator is a variation of *k*-fold which returns first *k*-fold as train set and the (*k* +1)th fold as the test set. Differently from standard cross-validation methods, here the successive training sets are super-sets of those that come before them, it also adds all surplus data to the first training partition which is always used to train the model. In our training data set, we have four months of data (the first and last months are not complete) and so we have split the data to four partitions.

8. Results

The goal of this work was to find the best performing model for water quality data, and check whether machine learning models are more accurate than logistic regression. This section presents the results obtained by running experiments to trained models for each algorithm. Tables 3 and 4 represent the final results of the experiment. The models are ordered according to the highest F1 score. Each model was the best performing model in the one variety group of models. For example, the SVM algorithm represented here, outperformed many other SVM models with different kernel and cost functions. On previous experiments we had presented at (Muharemi et al., 2018a), the

Table 4. Models performance using only feature importance.

Classifier	F1 score
SVM	0.36
DNN	0.06
LSTM	0.146
RNN	0.1
LogRegression	0.44
Simple NN	0.32

best model resulted to be the statistical model logistic regression where we considered interaction terms, based on the domain knowledge. A decrease of one pH unit is accompanied by an increase in Redox Potential, and there is a slight impact on the pH of pure water, it decreases as the temperature increases (Rodkey, 1959). From prior researches, in water quality, we see how water factors impact each other, and we can use them to build the model. We used this prior knowledge to add interaction terms in the logistic regression algorithm. Interaction terms played a very important role in achieving the best F1 score and winning the Industrial Challenge GECCO Competition 2017. When the interaction terms have been added to logistic regression we obtained a much more promising result than when we removed correlated features. Inspired by related works (mentioned in Section 2), we extended our experiment by using machine learning algorithms. The results show that machine learning algorithms perform well when evaluated with time-series cross-validator (see Table 3).

We also used an additional unseen test set which allows us to compare different models in an unbiased way, by basing the comparisons in data that were not use in any part of the training/hyperparameter selection process. Table 4 shows that the results are far more different than the one from the Table 3. This additional test set was provided after the GECCO 2017 competition was closed. Many machine learning classification algorithms assume that the target classes share similar prior probabilities and misclassification costs. However, this is often not the case in the real world. This unseen data set shows how bad the results can be when using highly imbalanced data sets. The results show that all algorithms are vulnerable although SVM, ANN and logistic regressions tend to be a little less vulnerable while DNN, RNN and LSTM are very vulnerable. One popular approach to solving the imbalanced data set problem is to resample the training set. Nguyen and Logofătu (2018) work achieved similar results using Tree Ensemble algorithm. However, few studies in the past have considered resampling algorithms on data sets with high dimensionality (Yap et al., 2014).

9. Conclusion

In this work, we presented approaches for event detection on water quality time series data. This work represents a case study and its aim is to find the best model on anomaly detection on water quality systems. Time series are analysed using statistical algorithms for a long time. Today machine learning algorithms are very popular on performing very well on time series data sets. Our experiment shows the weakness of machine learning algorithms when applied to a highly imbalanced data set. The previous work Muharemi et al. (2018a) has shown that SVM do not perform well on this time series

data set, and this happens due to not scaling the data. Here we can notice that scaling has improved the prediction while using SVM classifier. 'All models are wrong; some models are useful' (attributed to George Box). Since imbalanced data sets are found in so many different domains of application, it is highly desirable to find a more universal approach to the problem.

Notes

1. http://gecco-2017.sigevo.org/index.html/HomePage
2. https://www.svds.com/learning-imbalanced-classes/

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

Fitore Muharemi holds a M.Sc. degree in Computer Science from Frankfurt University of Applied Sciences in Frankfurt Am Main, Germany. She worked for a short time in the industry before starting her master studies. Fitore currently works as scientific researcher at the Faculty of Computer Science and Engineering, Frankfurt University of Applied Sciences. She does research in Data Mining, Artificial Neural Network and Artificial Intelligence.

Doina Logofătu holds a PhD degree in Computer Science from Babes-Bolyai University in Cluj-Napoca, Romania. She worked in the educational field as well in the industry side. She wrote several successful programming and mathematics books in Germany and Romania. At present she is professor for mathematics and computer science at Frankfurt University of Applied Sciences.

Florin Leon received a Ph.D. degree in computer science from the "Gheorghe Asachi" University of Iași, Romania in 2005, followed by a postdoctoral fellowship completed in 2007. In 2015, he defended his habilitation thesis. He has been a faculty member at the Department of Computers and Information Technology of the same university since 2005. In 2015, he became a Full Professor at the same department. He authored and co-authored more than 140 journal articles, book chapters and conference papers, and 12 books. He was a member in the guest editorial boards for three journal special issues, and he participated in 28 national and international research projects, two of which as principal investigator. He is currently a member of IEEE Systems, Man and Cybernetics Society: Computational Collective Intelligence Technical Community. His research interests include: artificial intelligence, machine learning, multiagent systems and software design.

References

Altman, E. I., Marco, G., & Varetto, F. (1994). Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the italian experience). *Journal of Banking & Finance*, *18* (3), 505–529.

Bekkar, M., Djemaa, H. K., & Alitouche, T. A. (2013). Evaluation measures for models assessment over imbalanced datasets. *Journal of Information Engineering and Applications*, *3*(10), 27–29.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory* (p. 144–152). ACM.

Bottenberg, R. A., & Ward, J. H. (1963). Applied multiple linear regression. Technical report, Personnel Research Lab Lackland AFB Tex.

Byer, D., & Carlson, K. H. (2005). Real-time detection of intentional chemical contamination in the distribution system. *Journal-American Water Works Association*, *97*(7).

Chandrasekaran, S., Freise, M., Stork, J., Rebolledo, M., & Bartz-Beielstein, T. (2017). Gecco 2017 industrial challenge: Monitoring of drinking water quality.

Che, Z., Purushotham, S., Cho, K., Sontag, D., & Liu, Y. (2018). Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, *8*(1), 6085.

García, S., Luengo, J., & Herrera, F. (2015). *Data preprocessing in data mining*. Springer.

Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., & Wierstra, D. (2015). Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*. *Vol. 112*. Springer.

Kang, G. K., Gao, J. Z., & Xie, G. (n.d.). Data-driven water quality analysis and prediction: A survey.

King, G., & Zeng, L. (2001). Logistic regression in rare events data. *Political analysis*, *9*(2), 137–163.

Mohammadpour, R., Shaharuddin, S., Chang, C. K., Zakaria, N. A., Ab Ghani, A., & Chan, N. W. (2015). Prediction of water quality index in constructed wetlands using support vector machine. *Environmental Science and Pollution Research*, *22*(8), 6208–6219.

Muharemi, F., Logofatu, D., Andersson, C., & Leon, F. (2018a). Approaches to building a detection model for water quality: A case study. *Modern Approaches for Intelligent Information and Database Systems* (p. 173–183). Springer.

Muharemi, F., Logofătu, D., & Leon, F. (2018b). *Review on general techniques and packages for data imputation in r on a real world dataset*. Springer.

Nguyen, M., & Logofătu, D. (2018). Applying tree ensemble to detect anomalies in real-world water composition dataset. *International Conference on Intelligent Data Engineering and Automated Learning* (pp. 429–438). Springer.

Rodkey, F. L. (1959). The effect of temperature on the oxidation-reduction potential of the diphosphopyridine nucleotide system. *The Journal of Biological Chemistry*, *234*(1), 188–190.

Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Fifteenth annual conference of the international speech communication association*.

Sapankevych, N. I., & Sankar, R. (2009). Time series prediction using support vector machines: A survey. *IEEE Computational Intelligence Magazine*, *4*(2)24–38.

Schafer, O. (2010). mix: Estimation/multiple imputation for mixed categorical and continuous data r package version 1.0-8.

Shipmon, D. T., Gurevitch, J. M., Piselli, P. M., & Edwards, S. T. (2017). Time series anomaly detection; detection of anomalous drops with limited features and sparse examples in noisy highly periodic data. *arXiv preprint arXiv:1708.03665*.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th international conference on Machine learning* (pp. 1096–1103). ACM.

Xiang, Y., & Jiang, L. (2009). Water quality prediction using ls-svm and particle swarm optimization. *Knowledge Discovery and Data Mining, 2009. WKDD 2009. Second International Workshop on* (pp. 900–904). IEEE.

Yang, X. (2009). Artificial neural networks. *Handbook of research on geoinformatics* (pp. 122–128). IGI Global.

Yap, B. W., Rani, K. A., Rahman, H. A. A., Fong, S., Khairudin, Z., & Abdullah, N. N. (2014). An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets. *Proceedings of the first international conference on advanced data and information engineering (DaEng-2013)* (pp. 13–22). Springer.

Zhang, M., & Sawchuk, A. A. (2011). A feature selection-based framework for human activity recognition using wearable multimodal sensors. *Proceedings of the 6th international conference on body area networks* (pp. 92–98).

Zhang, J., Zhu, X., Yue, Y., & Wong, P. W. (2017). A real-time anomaly detection algorithm/or water quality data using dual time-moving windows. *2017 Seventh international conference on innovative computing technology (INTECH)* (pp. 36–41). IEEE.