Accident Localization at the District Heating Network of Kaunas Region using Machine Learning

Mantas Bukauskas[a], Mantas Lukoševičius[a]

[a]*Faculty of Informatics, Kaunas University of Technology, Kaunas, Lithuania*

Abstract
Machine learning is constantly gaining popularity in real life applications. And one of them is prediction of various real-life events that depend on a huge number of factors that are hard to evaluate. In this article we describe the process of applying XGBoost — one of supervised machine learning methods — to help in prediction and localization of accidents in the district heating network of Kaunas region. We also investigate the importance of the different factors for these events.

Keywords
Supervised machine learning, xgboost, district heating, accident localization

1. Introduction

In everyday operation centralised district heating company "Kauno energija" is supervising more than 900 kilometers of district heating networks that provide heating and hot water to 118 891 customers (as of the end of 2017) in Kaunas region.

Every year pipe breakages in the district heating network occur. In most of these times the district heating services must be stopped for the customers. And due to the aged infrastructure, it is difficult to determine where did the accident happen. The only sign of accident is often a critical pressure drop or a frequent refill of the heating water in the system. There are a lot of cases when repair teams are excavating the area but do not find the accident and sometimes small accidents cannot be found and are compensated by system refill.

When an accident happens any information that would help to determine its location is helpful. Due to the amount and complexity of factors that cause accidents it is difficult to predict them. There are some complex solutions with thermodynamic and hydro-mechanics models in the market, which allow to calculate pipe breakage, but they are hard to use in everyday work and require a lot of investments and learning efforts.

Also, a Web application for network accident management TAVSIS was developed and it seemed like a good idea to integrate accident localization algorithms within that system. All things considered, it was decided to create a tool for the heating network supervising personal. And supervised machine learning algorithms seemed as an inexpensive and valid option to help in the process of pipe breakage localization.

2. Related works

We were able to find some similar studies where pipe break accidents were predicted using machine learning methods. In article [1] pipe breaks were predicted for water distribution network using pipe attributes and climatic data, since a monitoring network is not available [2]. The goal of the authors was to find pipes that can break soon to prioritize pipe replacements and repairs. Also different models are tested for best performance: RankBoost.B, Cox proportional hazard model, Naive Bayes, Logistic Regression and Artificial Neural Network. The provided results show that RankBoost.B is the most successful with AUC score of more that 0.85. In article [3] an ensemble of models are used to predict water utility pipeline condition. As input the authors use physical pipe attributes, environmental data, and operational factors and data obtained from physical models which are developed to understand the physical process of pipe deterioration [4, 5, 6].

Another article [7] describes pipe failure modelling for water distribution networks using boosted decision trees. To predict pipe failures authors use AdaBoost, RUSBoost, Random Forest, and Decision Tree models.

Although the mentioned articles describe similar methodology there are some major differences to our approach. First of all, we are targeting a district heating network. Secondly, we are using historic weather data as one of the inputs. And finally, we have a different goal - to locate pipe segment that have failed rather

IVUS 2020: Information Society and University Studies, 23 April 2020, KTU Santaka Valley, Kaunas, Lithuania
mantas.bukauskas@ktu.edu (M. Bukauskas);
mantas.lukosevicius@ktu.edu (M. Lukoševičius)

Figure 1: Illustration of a pipe breakage accident.

than analyse which pipes are most likely to fail in future.

As for a district heating network we were unable to find any related work. In article [8] authors use machine learning approach to detect faults by analysing temperature readings and some additional data from district heating substations. In article [9] authors use a completely different methodology - a deterministic – probabilistic structural integrity analysis to predict pipeline lifetime and probability of failure.

Our approach is more similar to real-time traffic accident localization approach described in articles: traffic accident prediction in the state of Utah (USA) [10]; predicting traffic accidents through heterogeneous urban data [11].

3. The data

3.1. Accident records

In this study we analyze pipe breakage accidents (see Fig. 1) that happened from the January 2013 to September 2019. There were 1 466 accidents in total that occurred during this time. We extracted these attributes: date and time of the accident occurrence (weekday, month, hour), geographic location. The accident data are collected using a GIS system by the company dispatchers that are supervising the district heating operations 24 hours a day in shifts. When an accident occurs a dispatcher marks its location on the pipe segment and this allowed us to use a spatial intersection with the pipe segments to determine how many accidents have occurred in each different segment.

3.2. Piping network

There were more than 49 200 of pipe network segments with a total length of 904.17 kilometers that are used to provide district heating network services to customers

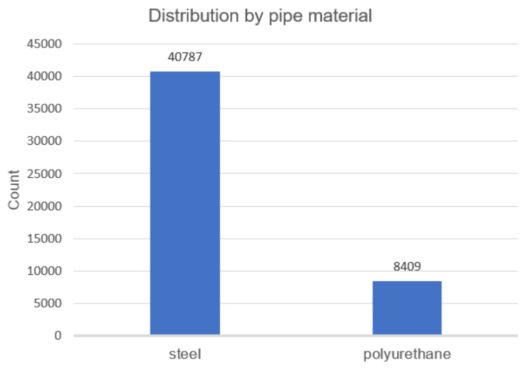


Figure 2: Distribution of pipe segment materials in Kaunas region

Table 1
Initial piping data summary for continuous values

Attribute	Mean	Median	Min	Max	Nulls
Year of installation	1995	1998	1963	2019	24500
Out. diameter, mm	165	100	25	1000	19580
Segment length, m	18.38	4.06	0.003	1118.74	0

in Kaunas region. We extracted these attributes from the pipe segments data table:

- type of anti-corrosion coating (bitumen varnish, no coating, unknown);
- type of isolation protection (unknown, plaster, tin);
- type of isolation (polyurethane, mineral wool, unknown);
- pipe material (distribution in the data is displayed in Fig. 2);
- type of pipe installation (distribution in the data is displayed in Fig. 3);
- year of the pipe installation;
- type of the heating water parameters (high, low);
- pipe line type (distribution in the data is displayed in Fig. 4);
- length of the pipe;
- pipe diameter.

A summary of the continuous values is presented in Table 1.

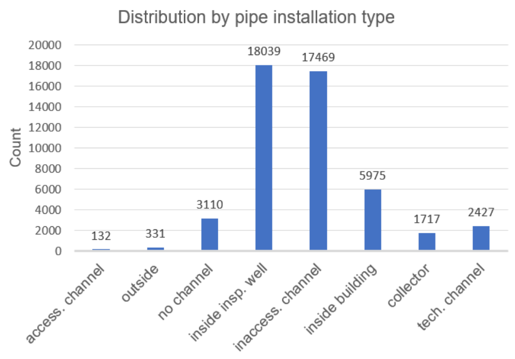


Figure 3: Distribution of pipe segment installation methods used in Kaunas region

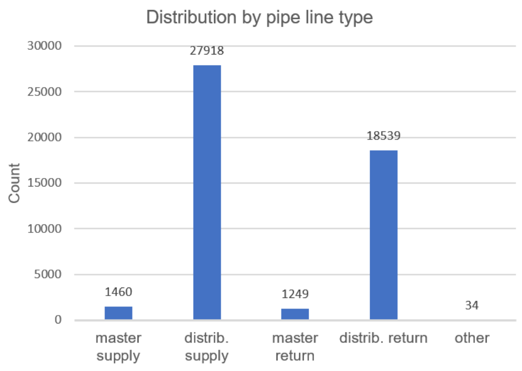


Figure 4: Distribution of pipe segments in different line types in Kaunas region

3.3. Weather data

Information about the weather conditions was extracted from archives on "Reliable prognosis" website[12]. We used one of Kaunas city weather stations. From the data set we used these attributes:

- temperature;
- atmospheric pressure;
- humidity;
- raining fact.

There were 19 724 records collected at 3-hour intervals during the analysed period January 2013 - October 2019. Data interpolation had to be used to get hourly records. Also, the raining fact was extracted from a human readable message rather than a numeric value.

4. Methodology

The process of our study was:

1. Find and collect necessary data from the data sources;
2. Analyse, identify, and extract useful data;
3. Select machine learning algorithms that would best fit the case;
4. Prepare training data for the model;
5. Split the prepared data set into training, validation and testing data sets;
6. Set machine learning model parameters;
7. Execute the learning process;
8. Validate the results;
9. Repeat steps 6 – 8 until expected results are reached;
10. Export the prepared model and test it on the testing data;
11. Deploy the prepared model to production.

4.1. Data preparation

Probably the most complicated part of this study was data preparation. We had to connect different sources of information to one — training data set which can be used by the XGBoost machine learning framework. All the processing and data analysis was performed using these tools:

- ArcGIS Pro software was used to manage geographic data and perform initial analysis;
- Jupyter Notebook software was used to develop and run all the process and share the results;
- Scikit-learn machine learning framework was used to prepare training data;
- XGBoost gradient boosting framework was used for model preparation.

There were some attributes that were dropped as they were considered unimportant. These were attributes with none or very few values, or irrelevant fields: who edited the data, when the last edit was performed, etc. The quality of the remaining data was not perfect either, as some relevant attributes were missing. To mitigate the problem, we prepared the training data by using different techniques:

- Data interpolation to increase the frequency of the weather data to one-hour intervals;
- Spatial intersection to connect accidents to pipe segments, to find the missing pipe attributes from the connected segments;
- Calculation of mean values to fill the missing continuous values;

- Calculation of the most frequent values to fill the missing categorical values;
- One-hot encoding to transform the categorical values to numeric as the model cannot handle non-numeric values. This method creates a binary column for each category and returns a sparse matrix or a dense array (depending on the sparse parameter). This encoding is needed for feeding categorical data to many scikit-learn estimators, notably linear models and SVMs with the standard kernels[13].

Also we added some additional properties: month, day of week, hour of day, was it raining during the accident or not, the total count of accidents in the segment. We rounded time of accidents to hours to the lower side dropping any remaining minutes or seconds as there is always some delay between when the accident really happens and when it is noticed.

When all the data cleaning was finished, we joined the data set of pipe segments to the data set of accidents. We performed this operation by using spatial join method provided by Arcpy library which allowed us to join accident record to the nearest pipe segment within less than 10 meter distance. Next, due to data imbalance as described in section 4.5 we generated 5 times as many negative samples as we have accident records. We used a negative sample selection technique [10]:

- Randomly select an accident record from the positive examples;
- Randomly alter: the pipe segment, the hour of the day, or the day of the year;
- If the new sample is not within the accident records, add it to the list of negative samples;
- Repeat until we have 5 times as many negative samples as positive.

This allowed us to work with a relatively low amount of data (in total 6 619 pipe segment in time records).

After that we connected the weather data to all these records. The final data set contained these attributes: count of accidents in segment, segment length, year of segment installation, segment diameter, weather temperature, weather humidity, raining state, atmospheric pressure, segment material, type of segment insulation, type of segment installation, subtype of the segment, hour of sample, weekday of sample, month of sample.

Finally, after the training data was prepared, it was split into two parts: 70 % for training and 30 % for testing. To make sure that positive (accident happened) and negative (accident did not happened) samples are distributed in equal rates for both data sets. For this we had to set the stratify parameter in the scikit-learn library for the output column indicating if the accident happened or not.

4.2. Model

As we mentioned before, our goal is to find most vulnerable pipe segments. We had to create a model which would be able to predict the probability of an accident in all piping segments at given situation. And by sorting these probabilities from highest to lowest we would be able to provide district heating network dispatcher with information which pipe segments are most vulnerable to having accident with current conditions: time properties (month, day of week, hour), weather conditions (temperature, humidity, atmospheric pressure, raining conditions) and pipe segment properties. By identifying the vulnerable pipe segments after the occurrence of the accident, the dispatcher can send repair teams to investigate them. It is important to mention that we do not really care how high or low the probability is in absolute value, because we already know that an accident has happened but we do not know where exactly.

In this study we used a decision tree ensemble based on gradient boosting algorithm called XGBoost. It was developed as a research project at the University of Washington. Since its release in 2016 it quickly gained popularity, won numerous Kaggle challenges, and is used in real-life applications. It is available as Open Source project and is actively developed by a community of data scientists. The XGBoost algorithm is based on gradient tree boosting model with additional regularization term which helps to smooth the final learnt weights to avoid over-fitting [14]. The regularized objective, or loss function, can be described as

$$L(\Phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k), \qquad (1)$$

where l is a differentiable convex loss function that measures the difference between the prediction $\hat{y}_i$ and the target y_i. The second term Ω penalizes the complexity of the model. Intuitively, the regularized objective will tend to select a model employing simple and predictive functions. This loss function can be integrated into the split criterion of decision trees leading to a pre-pruning strategy.

Furthermore, randomization techniques are also implemented in XGBoost both to reduce over-fitting and to increase the training speed.

There are multiple parameters that were tuned to get the best results (as described in XGBoost documentation[15]):

- max_depth – the maximum depth of the tree. Increasing this value will make the model more complex and more likely to over-fit.
- min_child_weight – the minimum sum of instance weight (Hessian) needed in a child. If the tree partition step results in a leaf node with the sum of instance weight less than the min_child_weight, then the building process will give up further partitioning. In a linear regression task, this simply corresponds to minimum number of instances needed to be in each node. The larger min_child_weight is, the more conservative the algorithm will be.
- eval_metric – type of evaluation metrics for validation data, a default metric will be assigned according to objective (RMSE for regression, and error for classification, mean average precision for ranking).
- objective – learning task and the corresponding learning objective.
- eta - step size shrinkage used in update to prevents over-fitting. After each boosting step, we can directly get the weights of the new features, and eta shrinks the feature weights to make the boosting process more conservative.
- early_stopping_rounds – activates early stopping. Validation metric needs to improve at least once in every early_stopping_rounds round(s) to continue training.
- num_boost_round – number of boosting iterations.

Weused this model from a Python API. These final parameters for model were chosen by hand while testing for the best results:

- max_depth = 6,
- min_child_weight = 5.0,
- eval_metric = 'auc',
- objective = 'binary:logistic',
- eta = 0.5.

4.3. Training

When the data was prepared, machine training was an easy step. After few try-outs we were able to get a model with 88.88 % of AUC (area under the curve) rating. It took only 13 epochs to reach this value, but the training kept going 50 epochs to determine that the value does not further increase.

4.4. Evaluation

One of the advantages of decision tree-based models is that it allows us to trace the key factors of how the decision was made. There are multiple characteristics that determine how well the model performs. The most important of them are:

Accuracy Accuracy explicitly takes into account the classification of negatives, and is expressible both as a weighted average of Precision and Inverse Precision and as a weighted average of Recall and Inverse Recall [16, 17, 18]. For us it shows the rate of correctly predicted result (accident or no accident):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{2}$$

where:

TP = True Positives;

TN = True Negatives;

FP = False Positives;

FN = False Negatives.

Precision It denotes the proportion of predicted positive cases that are correctly real positives. This is what Machine Learning, Data Mining and Information Retrieval focus on, but it is totally ignored in ROC analysis. It can however analogously be called True Positive Accuracy, being a measure of accuracy of predicted positives in contrast with the rate of discovery of real positives [17]. It shows how many accidents were predicted correctly compared to all predicted accidents

$$Precision = \frac{TP}{TP + FP}. \tag{3}$$

Recall It is the proportion of Real Positive cases that are correctly Predicted Positive. Recall has been shown to have a major weight in predicting the success of word alignment. In a medical context Recall is moreover regarded as primary, as the aim is to identify all Real Positive cases, and it is also one of the legs on which ROC analysis stands. In this context it is referred to as True Positive Rate [17]. For us it shows how many accidents were predicted correctly from all the occured accidents.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

ROC curve (Receiver Operating Characteristic curve) It is a graph showing the performance of a classification model at all classification thresholds. The curve plots two parameters [17]:

- True Positive Rate;
- False Positive Rate.

A ROC curve plots true positive rate vs. false positive rate at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. A perfect classifier will score in the top left-hand corner (False Positive Rate=0, True Positive Rate=100 %). A worst-case classifier will score in the bottom right hand corner (False Positive Rate=100 %, True Positive Rate=0). A random classifier would be expected to score somewhere along the positive diagonal (True Positive Rate = False Positive Rate) since the model will throw up positive and negative examples at the same rate [17].

AUC (area under the curve) The area under such a multipoint curve is thus of some value, but the optimum in practice is the area under the simple trapezoid [17]. As shown in Fig. 5, the main diagonal represents chance with parallel isocost lines representing equal cost-performance. Points above the diagonal represent performance better than chance and those below - worse than chance. For a single good (dotted green) system, AUC is area under the curve (trapezoid between the green line and x = [0,1]). The perverse (dashed red) system shown is the same (good) system with class labels reversed [17].

F1-score F-measure is defined as a harmonic mean of precision P and recall R [19]

$$F_1 = \frac{2PR}{P + R}. \tag{5}$$

4.5. Issues

During this research we met some issues that had to be overcome. Some of them are described below.

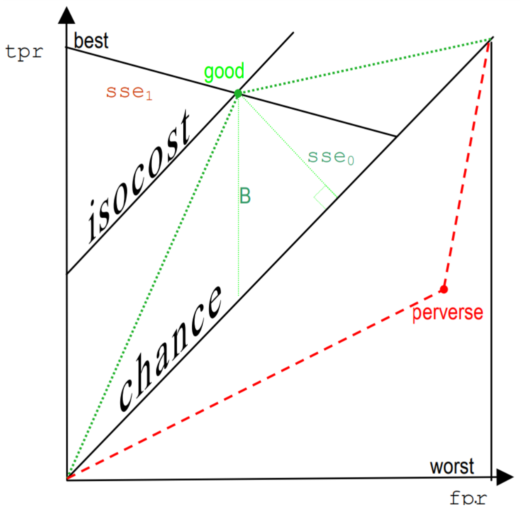


Figure 5: Illustration of ROC analysis[17]

Missing or incomplete data After the initial run only 30 % of positive samples was valid for processing as data was incomplete. Important features were missing such as pipe diameter, year of installation, material, etc. And that was a problem because we already had data imbalance issue with only 1 466 positive samples (pipe segments with registered accidents). One of the methods we used to calculate some missing values was spatial intersection. We have intersected pipes with each other and copied values from connected pipes assuming that connected pipes have the same parameters. Also we filled missing values by replacing them with most frequent values for categorical values and mean values for continuous values. Finally, 1 104 out of 1 466 of registered accidents were successfully used in the model.

Data imbalance Every year has at least 8 760 hours and every hour we have more than 49 000 of pipe segments that would make more than 429 million records every year. And we have only 1 466 accidents registered since 2013 January to 2019 October. If we would use data with this rate of positive and negative samples, the model would not be able to predict any of accidents. As by predicting that accidents will not happen at all it would be almost always right. To mitigate the problem we used negative sample selection technique as described in Section 4.1.

Non-linear factors of the accidents One of the problems of real-life event prediction is that there are many unknown and immeasurable factors that cause them to occur. There are many factors that might be

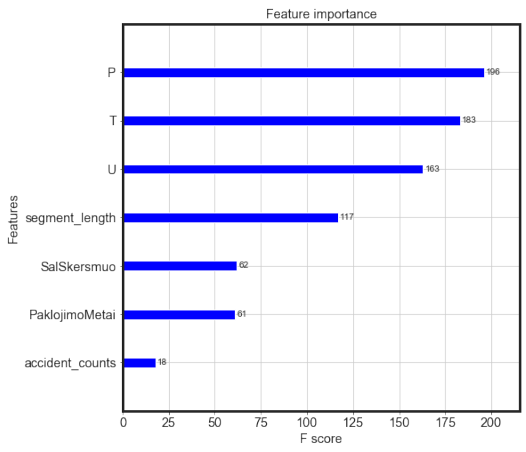


Figure 6: The most important features while constructing boosted decision trees according to the XGBoost library.

important, but we cannot determine them, or it would require a lot of effort to collect and provide them to the model. Such factors can be faults during the pipe production, transportation or installation, human errors, pipe environment conditions like soil, humidity, temperature changes, materials used in pipe production, electrical conductivity, chemical environment, etc.

5. Results

XGBoost library has a method to plot the most important features. Importance is a score that indicates how useful or valuable each feature was in the construction of the boosted decision trees within the model. The more an attribute is used to make key decisions with decision trees, the higher its relative importance. As shown in Fig. 6, most important features for district heating network accident prediction are weather pressure `P`, temperature `T`, humidity `U`, segment length `segment_length`, diameter `SalSkersmuo` and year of installation `PaklojimoMetai`.

Also as shown in Fig. 7, according to ROC curve our model performs well compared to random guess. Of course, we have to keep in mind that these parameters are provided for our data set where we limited our positive and negative sample rate to mitigate data imbalance. XGBoost classifier predicts probability of accidents between 0 and 1. By default, it has a threshold set to 0.5, meaning that probability higher than 0.5 will yield positive result and less than 0.5 - negative.

With probability threshold set to 0.16 we get these results:

- Test Accuracy: 80.31 %

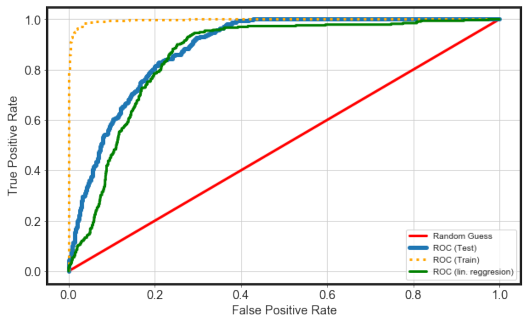


Figure 7: Receiver operating characteristic (ROC) curve

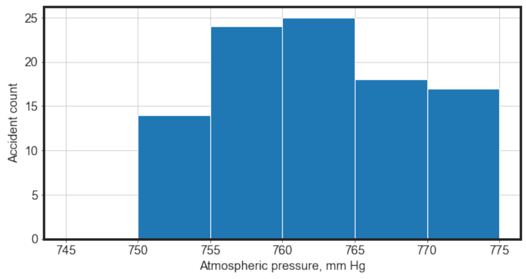


Figure 8: Histogram of atmospheric pressure vs. accident count

- Test Precision: 44.92 %
- Test Recall: 80.06 %
- Test F1: 57.55 %

But for our approach we care only about the highest probability with given parameters. Because we already know that an accident happened but we just do not know where exactly. That makes even relatively low probabilities valuable to us as it is additional information that can help us to find the accident locations. And our user - piping network dispatcher can use this probability to decide where it most likely have happened even if probability is relatively low. To compare our model performance we use AUC score. For XGBoost we get score of 0.868 as linear regression model reaches AUC score of 0.857. It can also be seen in ROC curve (see Fig. 7).

Also, we plotted some histograms with the most important features to see if they have any connection to pipe breakage.

As shown in Fig. 8, more accidents tend to happen when atmospheric pressure is between 755 and 765 mm Hg.

As shown in Fig. 9, accidents are more likely to happen during warm temperatures when heating service

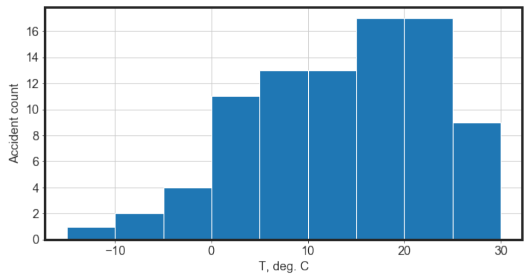


Figure 9: Histogram of temperature vs. accident count

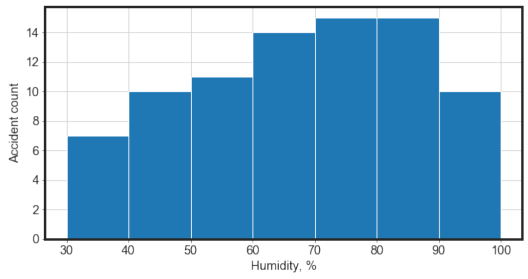


Figure 10: Histogram of humidity vs. accident count

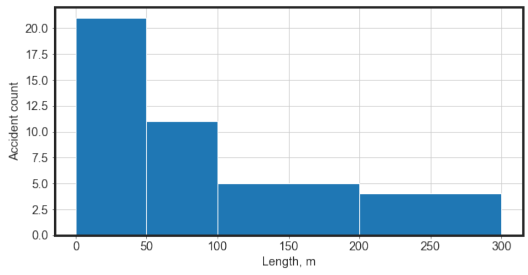


Figure 11: Pipe segment length vs. accident count

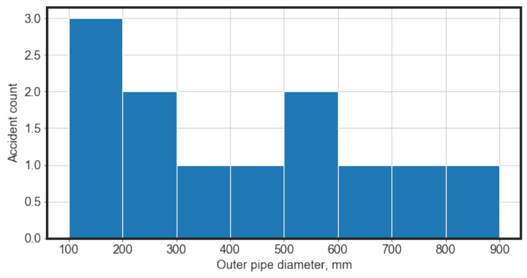
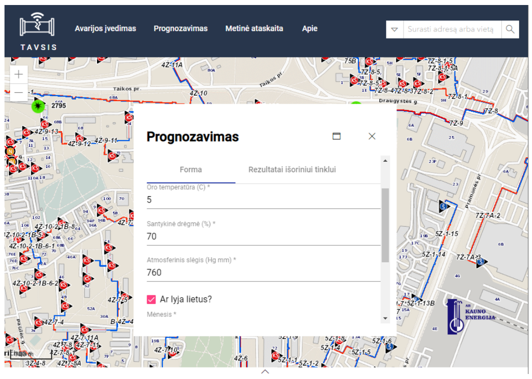


Figure 12: Pipe diameter vs. accident count



Prognozavimas

Oro temperatūra (C) *
5

Santykinė drėgmė (%) *
70

Atmosferinis slėgis (Hg mm) *
760

☑ Ar lyja lietus?

Mėnesis *

Figure 13: Accident prediction form within TAVSIS web application

is provided only for hot water. It can be explained as in summer hydraulic testing is performed.

As shown in Fig. 10, more accidents occur when humidity is higher.

As shown in Fig. 11, more accidents occur in short segments more often. It can be explained as new short segments are inserted during pipe repair.

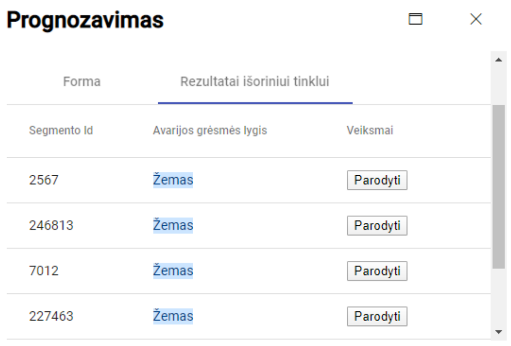
As shown in Fig. 12, accidents happens more often in pipes with smaller diameters.

6. Deployment

We have successfully trained a machine learning model using XGBoost framework, but the work was only halfway done, because this model had to be transferred to a production environment and used from a Web application. At first it seemed that it is a good idea to use a new framework from Microsoft called ML.Net. They declared that it is easy to integrate any machine learning model that can be converted to ONNX (Open Neural Network eXchange standard) model, and XGBoost was convertible, but after a lot of hours spent trying to launch the solution it was clear that XGBoost model was not fully supported and will not work. So a simpler approach was used – the model was published using a Flask RESTful server library and published through a reverse proxy on IIS. Occam's razor principle was proven right once again.

As shown in Fig. 13, we have created accidents prediction form within TAVSIS web application. It allows district heating network dispatcher to run trained model by filling form values: temperature, humidity, atmospheric pressure, raining conditions, month, weekday and hour.

As shown in Fig. 14, after submitting values to the accident prediction form user gets accident prognosis results. It is a list of pipe segments with accident possibility level (low, medium, high). User can zoom in to selected a pipe segment and get its location in the interactive map. This allows him to send a repair team to check if the pipe segment actually broke.


Prognozavimas

Forma | Rezultatai išoriniui tinklui

Segmento Id	Avarijos grėsmės lygis	Veiksmai
2567	Žemas	Parodyti
246813	Žemas	Parodyti
7012	Žemas	Parodyti
227463	Žemas	Parodyti

Figure 14: Accident prognosis results

7. Future work

Now we are working on applying different machine learning models and comparison of their characteristics to determine the best available model that would fit our needs. Also, we are evaluating how well the model performs in real-life applications.

In addition, we are also planning to collect more accident data from different district heating companies. Finally, we are considering applying this algorithm to different fields where piping networks are used, like water supply facilities, sewerage systems, etc.

8. Conclusions

This paper describes an ongoing study to apply supervised machine learning algorithms to help localize pipe breakage accidents in district heating network of Kaunas region. XGBoost model is trained and used to predict pipe breakages at the given time and weather conditions. And by sorting predicted probabilities from highest to lowest we are getting list of most vulnerable pipe segments. This allows us to provide this information to network dispatcher inside GIS web application and help him to determine the location of the accident.

References

[1] R. Wang, W. Dong, Y. Wang, K. Tang, X. Yao, Pipe failure prediction: A data mining method, in: 2013 IEEE 29th International Conference on Data Engineering (ICDE), IEEE, 2013, pp. 1208–1218.

[2] R. Giuliano, F. Mazzenga, A. Vizzarri, Satellite-based capillary 5g-mmtc networks for environmental applications, IEEE Aerospace and Electronic Systems Magazine 34 (2019) 40–48.

[3] F. Shi, Z. Liu, E. Li, Prediction of pipe performance with ensemble machine learning based approaches, in: 2017 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), IEEE, 2017, pp. 408–414.

[4] Y. Kleiner, B. Rajani, Comprehensive review of structural deterioration of water mains: statistical models, Urban water 3 (2001) 131–150.

[5] G. Capizzi, G. L. Sciuto, P. Monforte, C. Napoli, Cascade feed forward neural network-based model for air pollutants evaluation of single monitoring stations in urban areas, International Journal of Electronics and Telecommunications 61 (2015) 327–332.

[6] F. Beritelli, G. Capizzi, G. Lo Sciuto, C. Napoli, F. Scaglione, Rainfall estimation based on the intensity of the received signal in a lte/4g mobile terminal by using a probabilistic neural network, IEEE Access 6 (2018) 30865–30873.

[7] D. Winkler, M. Haltmeier, M. Kleidorfer, W. Rauch, F. Tscheikner-Gratl, Pipe failure modelling for water distribution networks using boosted decision trees, Structure and Infrastructure Engineering 14 (2018) 1402–1411.

[8] S. Månsson, P.-O. J. Kallioniemi, K. Sernhed, M. Thern, A machine learning approach to fault detection in district heating substations, Energy Procedia 149 (2018) 226–235.

[9] M. Valinčius, I. Žutautaitė, G. Dundulis, S. Rimkevičius, R. Janulionis, R. Bakas, Integrated assessment of failure probability of the district heating network, Reliability Engineering & System Safety 133 (2015) 314–322.

[10] D. Wilson, Using machine learning to predict car accident risk, 2018, https://medium.com/geoai/using-machine-learning-to-predict-car-accident-risk-4d92c91a7d57. Medium GeoAI publication.

[11] Z. Yuan, X. Zhou, T. Yang, J. Tamerius, R. Mantilla, Predicting traffic accidents through heterogeneous urban data: A case study, in: Proceedings of the 6th International Workshop on Urban Computing (UrbComp 2017), Halifax, NS, Canada, volume 14, 2017.

[12] Weather archive in Kaunas, last accessed May 2020. URL: https://rp5.ru/Weather_archive_in_Kaunas.

[13] OneHotEncoder, last accessed May 2020. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html.

[14] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 785–794. URL: https://doi.org/10.1145/2939672.2939785. doi:10.1145/2939672.2939785.

[15] Xgboost parameters, last accessed May 2020. URL: https://xgboost.readthedocs.io/en/latest/parameter.html.

[16] C. Napoli, E. Tramontana, M. Wozniak, Enhancing environmental surveillance against organised crime with radial basis neural networks, in: 2015 IEEE Symposium Series on Computational Intelligence, IEEE, 2015, pp. 1476–1483.

[17] D. Powers, Ailab, Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation, J. Mach. Learn. Technol 2 (2011) 2229–3981. doi:10.9735/2229-3981.

[18] F. Beritelli, G. Capizzi, G. Lo Sciuto, C. Napoli, M. Woźniak, A novel training method to preserve generalization of rbpnn classifiers applied to ecg signals diagnosis, Neural Networks 108 (2018) 331–338.

[19] Y. Sasaki, The truth of the f-measure, Teach Tutor Mater (2007).