Utilization of Machine Learning to Detect Sudden Water Leakage for Smart Water Meter

Jan Merta
Department of Process Control
FEI, University of Pardubice
Pardubice, Czech Republic
E-mail: Jan.Merta@upce.cz

Jan Fikejz
Department of Software Technologies
FEI, University of Pardubice
Pardubice, Czech Republic
E-mail: Jan.Fikejz@upce.cz

Abstract—**This article deals with the use of machine learning to detect sudden water leakage. A smart water meter, which enables monitoring the water consumption of the observed object, is used as the source of input data. Based on these data and their analysis, a symbolic regression, which must know not only the input parameters but also the structure of the model, was finally used to build the model. After finding a suitable function and standard deviation from the model, it is possible to set the required sensitivity and thereby detect anomalous states of water consumption in monitored time windows. Since the smart water meter also has a ball valve, if a sudden water leakage is detected, the water meter can autonomously close the main supply and thus avoid extensive damage.**

Keywords— Smart water meter, water leak, machine learning, symbolic regression

I. Introduction

Nowadays, we witness the dynamic development of industry 4.0 that can be found in all areas. One of the industry's 4.0 components is the Internet of Things (*IoT*), which is mainly used for consumer, business and infrastructure applications. Besides the Internet of Things, it is also possible to see *EIoT* (Enterprise Internet of Things), which is mainly used in business solutions. Currently, experts estimate that the Internet of Things will include approximately 30 billion devices in 2020, and the market value is estimated to be $ 80 billion [1].

Within this area, for example, smart buildings can be built to monitor key systems and units. One of the important monitored substances is water, for which the most monitored are its quality or consumption. Water is nowadays an increasingly discussed commodity, as it is one of vital substances. One of the reasons why water attracts ever more attention is its depletion. It mainly involves a significant drop in groundwater [2]. As a result, many supporting activities come not only from the state to address this issue in the future. However, the fact remains that water is and will be a valuable commodity, and one of the areas that is particularly important for end customers is the monitoring of its consumption and early detection of unwanted leakage. On average, around 20% loss of the total water distribution [3] is reported worldwide.

Sudden leakage of water can cause considerable damage in the short term, especially to equipment adjacent to the point of leak. On the other hand, such leakage is relatively easy to detect. Another significant problem can be a small but long-term and continuous leakage. In these cases, there is not necessarily a great damage to property, but rather in higher or additional water costs. It is precisely these long and continuous water leaks that should be detected and, in the event of their occurrence, inform about them in good time and adequately respond to them. One way to detect such leaks is to use machine learning, which, based on the model found, can detect these events in a timely manner.

II. State of Art

In the area of water monitoring, a number of entities or organizations already currently operate, particularly in the area of surveillance. This is, for example, a design of real-time automated reading system using *IoT* to be deployed in smart cities [4]. These systems use a system of measuring sensors and communicate via, for example, *ZigBee* wireless communication technology. Data is most often stored on cloud servers for more detailed data processing.

Other systems, for example, monitor grid pressure [4, 6]. With these systems, water distributors can remotely monitor network status and detect sudden leakages early on both the backbone network and end nodes such as city hydrants.

On the basis of long-term monitoring of water distribution networks, it is possible to, for example, determine the age of the network and to detect the no longer adequate condition of individual parts of the pipeline. In these cases, it is most often a continuous increase in consumption at the same measured water flow [7].

Most systems, however, primarily aim at the industrial use of distribution networks and monitoring the consumption of larger agglomerations as one of the smart city subsystems.

III. Smart Water Meter

The use of a smart water meter for monitoring water consumption with the possibility of automatic closing of the main inlet is an additional technical solution with application ranging from flats and private houses to administrative buildings or educational institutions. Especially in buildings with a higher number of persons, more complex water distribution systems with higher number of end-points are built, such as sanitary facilities.

These end-points are the most common source of water leakage problems. Excessive use leads to higher wear and thus to an earlier potential failure. This often leads either to sudden leakage or long-term water leakage, which in both cases entails additional costs [8].

A. *Hardware*

The basic part of the smart water meter is shown in Figure 1 and consists of four parts:

- **Central control unit**- mini-computer expanded by a real-time unit (RTC) using the standard Linux *Raspbian* environment. In this Linux environment there is implemented software, which mainly provides:
    - reception of information about flow rate from the water meter,
    - communication within the application interface via a wireless network,
    - periodic evaluation of the water flow based on established rules according to application logic,
    - two-way valve control.
- **Pulse Water Meter** - Provides information on the intensity and continuity of water flow. In general, any pulse water meter can be used, even with any sensitivity (number of pulses per liter).
- **Two-way ball valve -** based on the request from the control system (both external from the user and internal based on the evaluation of the autonomous mode) controls the opening or closing of the water flow.
- **Backup battery -** provides a standby power supply in the event of a power outage, for greater stability when the main power supply has been dropped.

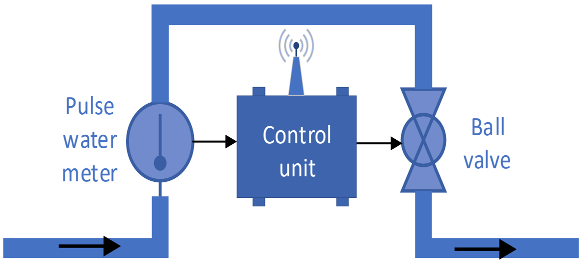

Fig. 1. Basic concept of a smart water meter.

B. *Application interface*

The whole system can be divided into three layers:

- measurement-water meter layer,
- communication layer,
- user layer.

The overall concept of the smart water meter system is illustrated in Figure 2.

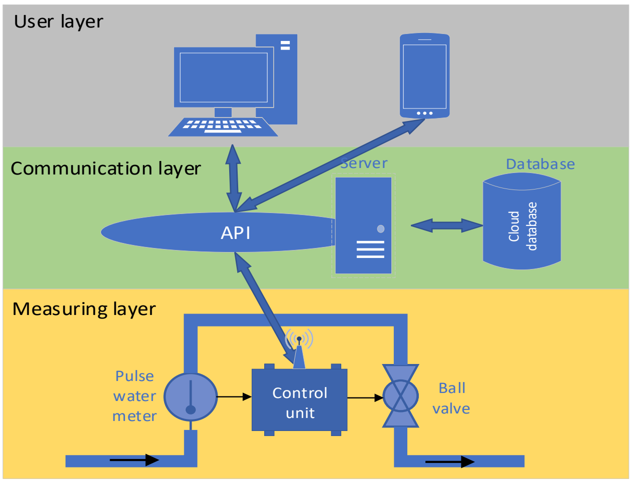

Fig. 2. Overall concept of the smart water meter system.

In order to communicate with the smart water meter is used a suitable application interface (API). For this case, the REST (Representational State Transfer) architecture was chosen. Application interface is implemented as a web service and consists of a set of methods that provide basic communication means between:

- user layer and database,
- water meter and database.

IV. Anomaly Detection

Anomaly detection deals with searching for unusual observations (anomalies) in data often referred to as outlier detection [10]. The aim of the method is to detect (or possibly remove) an abnormal data point that differs considerably from most other data points. More precisely, the points that do not correspond to the model based on the data being searched. These points are called outliers [9, 10]. Differentiation of outliers can be binary (is or is not an outlier) or "fuzzy-like", where scores can be calculated for each point to determine the extent to which it is an outlier.

A. *Outliers*

There is no universally accepted definition for these points [10]. In his publication [11], Hawkins describes outliers as follows: "An observation which deviates so much from other observations as to arouse suspected that it was generated by a different mechanism ".

Outlier detection is used for automatic signaling of failures and errors, notification of changes in system behavior, discovering suspicious behavior (for example, attackers on a computer network) and fraud detection [10, 12].

There are several different approaches for outlier detection. Supervised data methods in which normal (and often abnormal) data points are explicitly indicated. Then it is sufficient to teach the classifier to distinguish between normal and abnormal points using these labeled data [13].

Unsupervised methods use unlabeled data and are based on searching patterns in data. These include, for example, various statistical tests, clustering methods, etc. [13]. Model based methods look for a mathematical model describing the data and marks points above the model prediction boundary as outliers [12]. For these purposes, a method based on the standard deviation of model errors may be used against real data. The model error for a particular data point *pi* is calculated as the difference between the true value of *y* point p_i in the data and the functional value of the found equation at point p_i, equation 1.

$$error_{p_i} = y_{p_i} - f(x_{p_i}) \qquad (1)$$

Then the standard deviation $SD_{of\,an\,error}$ from all errors of all points in the dataset, describing how much the model differs from the real data, is calculated. The calculated value can then be used to separate abnormal points. We will consider data points whose difference between the real value and the predicted value is greater than the threshold, defined as *k* multiple of the standard deviation of an error of the equation 2 as outliers.

$$threshold = k \cdot SD_{of\_an\_error} \qquad (2)$$

We can substitute k with, for example, value 3 [8] according to 3-sigma rule [14]. This rule states that when data comes from normal distribution, 99.87% of data points will be located within three standard deviations from the average [14, 16]. Other authors recommend a stricter threshold of 2.5 or even 2 standard deviations from the average [17]. The choice of the standard deviation multiple depends on the specific problem and data [15]. In their article, the authors Leys et al. [15] even recommend using a different approach based on the absolute deviation from the median instead of the standard deviation (sensitive to extreme values).

Various machine learning methods, such as regression methods or neural networks, can be used to find the model. While neural networks are inherently hard to interpret (black boxes), regression can find a mathematical equation representing the data model. Classical regression is based on the search for model parameters (line slope in linear regression, polynomial coefficients in polynomial regression). The model structure (linear, quadratic, polynomial regression) is selected manually to best describe the character of modelled data. The need to select the structure of the model manually makes classical regression a method that (as opposed to the neural network) is hard to scale. This problem is solved by symbolic regression, which is often implemented through genetic programming.

V. Genetic Programming

Genetic programming (introduced by J. R. Koza) is an extension of genetic algorithms that can develop programs through evolution. Chromosomes are of variable lengths and have the form of syntactic trees whose vertices are either terminals (constants, variables, random numbers) or non-terminals (functions). For each problem, it is always necessary to select a suitable set of terminals and non-terminals. The variable length can cause rapid growth in the average length of trees in the population [18, 19].

Genetic programming has a similar sequence of steps as genetic algorithms (random initial population of individuals, evaluation of the fitness function of individuals, selection, creation of a new population of different reproduction and mutation methods). For example, when subtree crossover is exchanged between two individuals, their random subtrees are exchanged, during subtree mutation the random subtree of the individual is discarded and a new one is generated. The resulting syntactic tree represents the structure of the generated program [18, 19].

VI. Symbolic Regression

Symbolic regression with genetic programming is a supervised machine learning method that can find an equation (in the form of a syntactic tree) describing the given dataset. It optimizes model parameters as well as its structure through evolution. It is not necessary to know in advance whether the data describes a polynomial of the 2nd or 5th order. At the same time, it may not just be a polynomial, but the equation may contain functions such as sine, logarithm, etc. The set of non-terminals usually contains selected mathematical operations and functions (*, +, -, /, sin...) and the set of terminals contains constants and variables (2, 7, 6.78, π, e, ...). Fitness is usually defined as the sum of differences between expected outputs and outputs from the model [19].

VII. Model Search Data

Before searching for the model, which would reflect the variable behavior of the observed objects, it was necessary to prepare and analyze the input data on the consumption of the selected object. Data on the cumulative consumption from the water meter is stored in the database every hour via REST API. So, we used these data for a period of six months, Figure 3.

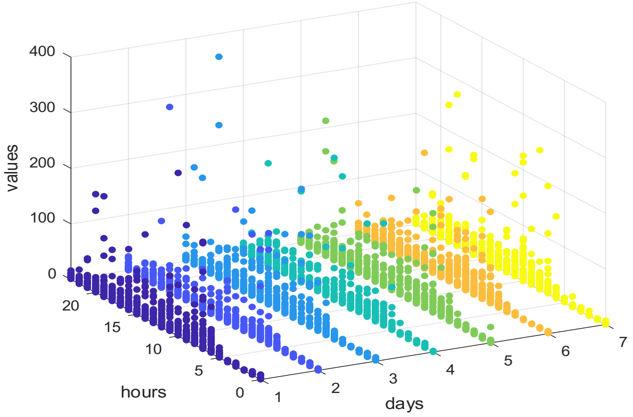

Fig. 3. Data in 3D model.

Since we wanted to look at the data by day of the week, each axis of the graph represents time (z axis), days of the week (x axis) and consumption in liter (y axis). Another angle of view, from which daily maxima are visible, can be obtained from the 2D model, Figure 4.

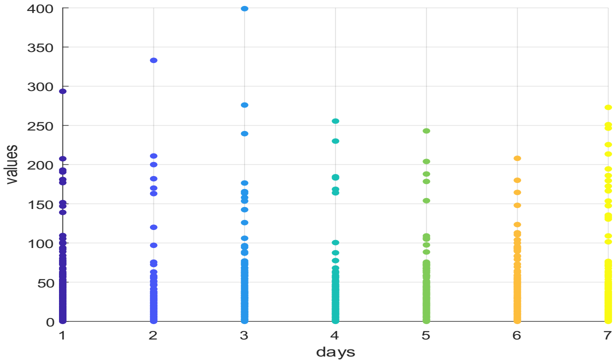

Fig. 4. Consumption data by day.

VIII. Building the Model

Due to the nature and variability of the data of the observed objects, a scalable machine learning method was chosen for modelling the water consumption system. Finally, the method of symbolic regression in combination with genetic programming was chosen to search for the model. The model was then searched for each day separately, which better captures reality.

You can simplify the procedure of searching for a model for a single day in three steps:

1. Filter out data for only one specific day of the week.
2. Find the model (equation) of a given day using symbolic regression.
3. Calculate the standard deviation of the model error relative to all data points.

After finding the model, using the k-sigma method (*k* multiple of the standard deviation) we can use offline learning in historical data to find points that are abnormal in terms of long-term trend.

If this difference is greater than the chosen *k* multiple of the standard deviation, this point can be declared unusual — outlier. Alternatively, it is possible to calculate how many times a given value exceeds the standard deviation in successive measurements.

For individual hours, more stringent and less stringent tolerance (another *k*) can be chosen.

The genetic programming algorithm for symbolic regression was run in all attempts with the options and parameters listed in Table 1.

TABLE I. ALGORITHM PARAMETERS

Maximum tree depth when generating an initial population	2
Initial population generation method	Grow method
Crossover method	Subtree crossover
Mutation method	Subtree mutation
Random Number Generator	Mersenne Twister
Elitism	1
Population size	20
Selection	Tournament selection
Set of terminals	Random real number (from -1 to 1), hour of measurement
Set of non-terminals	Multiplication (*) and addition (+)

The calculation of fitness of individuals consisted of the sum of the differences between the outputs from the model and expected values (sum of squared residuals) divided by the number of data points and ten percent of the size (number of vertices) of the syntactic tree equation (penalization of large trees), equation 3.

A. Results

For example, the data reflecting the 3rd day of the week, i.e. the Wednesday, the dependence of which is shown in the graph in Figure 5, can be considered a representative example of the input for finding the model.

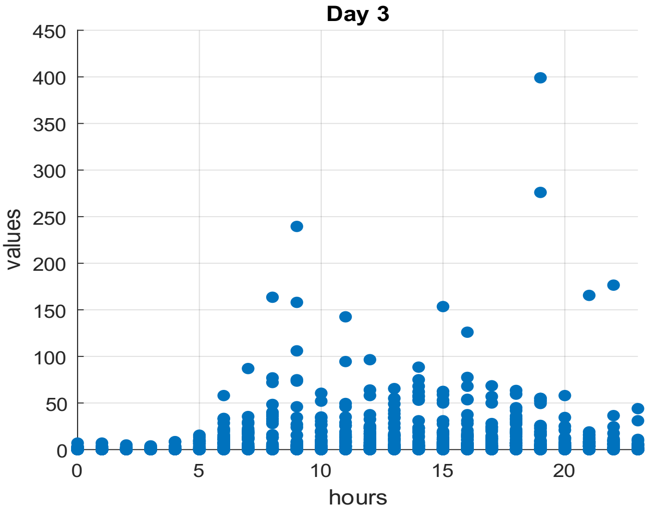

Fig. 5. Dependence of the measured values on the hour of measurement during the 3rd day.

After five hundred generations of the genetic algorithm, equation 3 was found on the basis of historical offline data and subsequent modification and it is then shown in Figure 6.

$$y = -0{,}0018x^3 - 0{,}0293x^2 + 1{,}8239x \qquad (3)$$

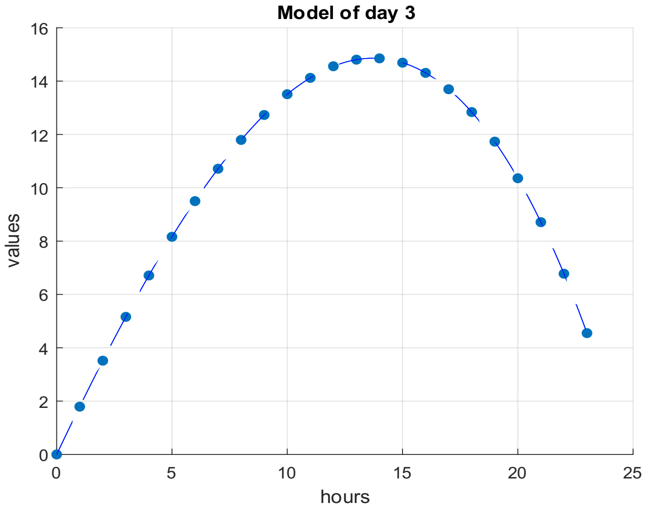

Fig. 6. Found model for Day 3.

The standard deviation of the model error is 24.21965 liters. Subsequently, a multiple of the standard deviation can be determined and thus set the sensitivity threshold and search for an abnormal state.

Individual models representing individual days can then be implemented within a smart water meter to detect non-standard water abstraction states that can represent unwanted water leaks in the observed building.

IX. Conclusion

Attention was primarily paid to the design of the model using machine learning. Information from a smart water meter, which at hourly intervals provides information on the cumulative water consumption of the monitored object, was used as input data for finding the model.

Due to the nature and variability of the data of the observed objects, a scalable machine learning method was chosen for modelling the water consumption system. Finally, the method of symbolic regression in combination with genetic programming was chosen to search for the model. The input data was divided into individual days and then a set of models and their standard deviations were found for each day. These models can then be used to detect non-standard water abstraction states, which may represent unwanted water leaks in the monitored object. By the multiplicity of the standard deviation, the sensitivity threshold for abnormal consumption values in different time windows can be set. The frequency of successive non-standard consumption values may also be considered further, which will be the subject of further research.

Acknowledgments

The work has been supported by the Funds of University of Pardubice (by project “SGS 2019 No: SGS_2019_021), Czech Republic. This support is very gratefully acknowledged.

References

[1] K. Mundle, Home Smart Home: Domesticating the Internet of Things. *Toptal.com* [online]., 2015 [cit. 2018-01-05]. https://www.toptal.com/designers/interactive/smart-home-domestic-internet-of-things

[2] The biggest problem today - water. *Prumyslovaekologie.cz* [online]. Průmyslová ekologie, 2017 [cit. 2018-01-05]. http://www.prumyslovaekologie.cz/Dokument/103217/nejvetsi-problem-soucasnosti-voda.aspx

[3] G. Moser, S. G. Paal, and I. F. C. Smith, Leak Detection of Water Supply Networks Using Error-Domain Model Falsification, *Journal of Computing in Civil Engineering*, vol. 32, no. 2, p. 04017077-, 2018.

[4] H. Ali, W. Y. Chew, F. Khan, and S. R. Weller, Design and implementation of an IoT assisted real-time ZigBee mesh WSN based AMR system for deployment in smart cities, *2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, pp. 264-270, 2017.

[5] Y. Gao, M. J. Brennan, and P. F. Joseph, A comparison of time delay estimators for the detection of leak noise signals in plastic water distribution pipes, *Journal of Sound and Vibration*, vol. 292, no. 3-5, pp. 552-570, 2006.

[6] G. Moser, S. G. Paal, and I. F. C. Smith, “Leak Detection of Water Supply Networks Using Error-Domain Model Falsification, *Journal of Computing in Civil Engineering*, vol. 32, no. 2, p. 04017077-, 2018.

[7] A. Rajeswaran, S. Narasimhan, and S. Narasimhan, “A graph partitioning algorithm for leak detection in water distribution networks, vol. 108, pp. 11-23, 2018.

[8] J. Fikejz and J. Rolecek, “Proposal of a smart water meter for detecting sudden water leakage, *in 2018 ELEKTRO*, 2018, pp. 1-4

[9] A. Zimek and a E. Schubert. Outlier Detection. LIU, Ling a M. Tamer ÖZSU, ed. *Encyclopedia of Database Systems* [online]. New York, NY: Springer New York, 2017, 2017-09-27, s. 1-5 [cit. 2019-02-23]. DOI: 10.1007/978-1-4899-7993-3_80719-1. ISBN 978-1-4899-7993-3. http://link.springer.com/10.1007/978-1-4899-7993-3_80719-1

[10] V. Hodge, and J. Austin. A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review* [online]. 2004, 22(2), 85-126 [cit. 2019-02-23]. DOI: 10.1007/s10462-004-4304-y. ISSN 0269-2821. http://link.springer.com/10.1007/s10462-004-4304-y

[11] M.D. Hawkins, D. M. *Identification of Outliers* [online]. Dordrecht: Springer Netherlands, 1980 [cit. 2019-02-23]. ISBN 978-94-015-3996-8.

[12] H.-P. Kriegel, P. Kroger, and A. Zimek. Outlier Detection Techniques. *Conference Tutorial at SIAM Data Mining Conference*, 2010. http://www.siam.org/meetings/sdm10/tutorial3.pdf [5]

[13] Ch. Aggarwal, Outlier analysis. New York: Springer, 2013. ISBN 9781461463962.

[14] L. Kazmier, *Schaum's outline of theory and problems of business statistics.* 3rd ed. New York: McGraw-Hill, c1996. ISBN 0070340269.

[15] Ch. Leys, O. Klein, P. Bernard and L. Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology* [online]. 2013, 49(4), 764-766 [cit. 2019-02-23]. DOI: 10.1016/j.jesp.2013.03.013. ISSN 00221031. https://linkinghub.elsevier.com/retrieve/pii/S0022103113000668

[16] D.C. Howell, (1998). *Statistical methods in human sciences*. New York: Wadsworth

[17] J. Miller,. (1991). Reaction time analysis with outlier exclusion: Bias varies with sample size. T*he Quarterly Journal of Experimental Psychology,* 43(4), 907–912, http://dx.doi.org/10.1080/14640749108400962.

[18] J. R. Koza ,Genetic programming: *On the programming of computers by means of natural selection.* Cambridge: Bradford Book, 1992. ISBN 0-262- 11170-5.

[19] P. Riccardo, W. B. Langdon and N. F. McPhee. A *field guide to genetic programming.* [S.l.: Lulu Press], 2008. ISBN 978-1-4092-0073-4