Research Article

O. Kainz*, E. Karpiel, R. Petija, M. Michalko, and F. Jakab

Non-standard situation detection in smart water metering

<https://doi.org/10.1515/comp-2020-0190>

Received Jan 27, 2020; accepted May 13, 2020

Abstract: In this paper an algorithm for detection of non-standard situations in smart water metering based on machine learning is designed. The main categories for non-standard situation or anomaly detection and two common methods for anomaly detection are analyzed. The proposed solution needs to fit the requirements for correct, efficient and real-time detection of non-standard situations in actual water consumption with minimal required consumer intervention to its operation. Moreover, a proposal to extend the original hardware solution is described and implemented to accommodate the needs of the detection algorithm. The final implemented and tested solution evaluates anomalies in water consumption for a given time in specific day and month using machine learning with a semi-supervised approach.

Keywords: anomaly detection, machine learning, non-standard situation detection, smart water metering, water consumption

1 Introduction

In recent years, the interest in smart homes has been on the rise. One of the possible reasons is the high availability of offered solutions at affordable prices, while providing simple installation. These devices offer control of common household appliances or even lighting. They can also provide the user with various up-to-date information, such as indoor temperature or air humidity, measured by sensors incorporated in such devices. The data can be sent either to the home server outside the external network or just stored on the network, in the so-called “cloud”, which the

user can access from any location. These can be accessed through a web interface, a mobile application, or any other solution supported by the device.

Users may be also interested in energy consumption. Most products are already manufactured to save energy (like water, electric energy, etc.). Many households have an analog water meter to measure the water consumption. These analog water meters contain dials on which the total water consumption is presented. The consumer alone would have to record these values and then compare the new ones to evaluate their current water consumption, *e.g.*, per hour.

Algorithms for image processing exist and can convert dials of water meters into digital form. In addition to digitization of these values, there are also approaches for other so-called heuristic processing. Through the usage of modern machine learning techniques, the non-standard situations may be automatically evaluated.

The detection of non-standard situations, or anomaly detection, according to Penttilä [1], represents the detection of anomalous data in a known data set. This term is closely related to both statistics and machine learning. Anomalies can be detected in various areas, such as banking (fraud detection in research by Dorj and Altangerel [2]), medicine (detection of non-standard values in health records, *e.g.*, method presented by Carvalho *et al.* [3]) or information technology to detect the potential critical system failures (Hodge and Austin [4]).

Machine learning techniques depend on the context in which they are to be applied, on the training data set and whether all types of non-standard situations are known. Detecting non-standard situation in water consumption by applying machine learning should be accurate enough to be considered a reliable and correct solution with minimal human intervention. The solution described in this paper is a component of the overall solution for metering of water consumption that can be easily enabled in small and medium households that utilize analog water meters. Next, we will focus only on the detection of non-standard situations from such measurements through our own algorithm.

*Corresponding Author: O. Kainz: DCI, TUKE;

Email: ondrej.kainz@tuke.sk

E. Karpiel: Siemens Healthineers; Email: eduard@karpiel.sk

R. Petija: DCI, TUKE; Email: rastislav.petija@tuke.sk

M. Michalko: DCI, TUKE; Email: miroslav.michalko@tuke.sk

F. Jakab: DCI, TUKE; Email: frantisek.jakab@tuke.sk

2 Categories for Non-standard Situation Detection

Based on research in [5] and [6] there are three main categories for anomaly detection: supervised, semi-supervised and unsupervised.

A non-standard situation or anomaly may be considered any event occurring out of standard boundaries. In our case it is based on the recorded actual water consumption, which is considered standard.

2.1 Supervised

Supervised detection assumes that the standard and anomalous data is labeled in the available training dataset. In this case, this is done by creating a prediction model for this data, which then compares them with other data and tries to determine whether they are anomalous or not. The possible issue in this case, as the authors mention in research [5], is a situation with a larger set of data, where part of it is unknown for a given training set. Then the model will not be able to distinguish whether a given data instance is or is not anomalous. Thus, the supervised detection is mainly utilized when all types of anomalies are known and their occurrence is uncommon.

2.2 Semi-supervised

The semi-supervised detection has, according to research in [6], an available set of training data with defined labels only for standard data. This, in contrast to the previous example, brings a wider range of applications since the created prediction model can distinguish whether a given data instance is normal or not based on the training data. If it is not normal, then it is anomalous. In this case, the set of possible anomalies can be unlimited.

2.3 Unsupervised

Unsupervised detection algorithms carry out learning by themselves and do not use a training data set. Authors in [5] reported this detection as the most commonly used technique. This technique assumes that anomalies occur less frequently than standard data in a given dataset, thus patterns and relations become apparent. However, if this assumption is not true, techniques based on this detection will contain many false anomalies.

3 Classification of methods for non-standard situation detection

According to research in [6] there are several methods for detection of non-standard situations based on classification, nearest neighbor or clustering. Other approaches include statistical and spectral methods. However, we primarily analyze two of them: methods of anomaly detection based on clustering and statistical methods of anomaly detection.

3.1 Clustering-based anomaly detection

Authors of [7] describe clustering as a grouping of observation into subgroups, where these groups are much more closely interrelated than if they were linked to observations from other groups. Consequently, it is possible to look for differences between the observations. As is described in [6] there are three different situations for anomaly detection for this method:

1. Anomalous data does not belong to any cluster.
2. Anomalous data is far from their closest cluster, which is the arithmetic mean of all points in a given cluster.
3. Anomalous data belongs to a small or sparse cluster.

The advantages of clustering-based anomaly detection include the ability to work in a partially supervised mode (can detect even unknown anomalies) and the ability to adapt techniques to different data types. Most algorithms are mainly focused on clustering and are not directly designed to detect anomalies; this can be the disadvantage when using clustering technique for the detection.

3.2 Statistical anomaly detection

Anomalies, in case of statistical anomaly detection methods, are according to [6] detected based on a statistical model. Thus, a data instance is or is not matched to the model. Data instances that are less likely to be generated by the statistical model are considered anomalous. Hodge and Austin [4] categorize statistical methods into four groups: proximity-based methods, parametric, non-parametric, and semi-parametric methods.

3.2.1 Proximity-based methods

These methods are simple to implement and have no assumptions of the data distribution model. On the other hand, the time consumption exponentially increases with respect to the number of the data, since the distance between all data is calculated.

The example of these methods is the k-NN algorithm, based on the closest neighbor that calculates the point distance from other points. If this distance is significantly larger, the point may be considered as anomalous. Euclidean or Mahalanobis distance may be used to calculate the distance.

As mentioned, with the increasing number of data, the time consumption of this algorithm increases exponentially. A possible solution is its optimization, which is described in detail in the research by Hodge and Austin [4].

3.2.2 Parametric methods

Parametric methods allow the statistical model to be applied very quickly to new data instances and hence are also suitable for large datasets, as described in [4]. Subsequently, they draw attention to a possible disadvantage by using a pre-selected distribution model that matches the data. However, not all data must fit into one particular model, leading to incorrect anomaly reports.

Chandola *et al.* in [6] present several examples of parametric anomaly detection methods, which are divided into methods based on: Gaussian model, regression model and parametric distribution mixture. Detailed analysis of techniques for these methods can be found in [4] and [6].

3.2.3 Non-parametric methods

Non-parametric methods are, according to [4], suitable for data that do not fit only in one model; they can be distributed randomly and thus it is not possible to have a pre-created model for them. Thus, non-parametric methods for detecting anomalies can be used for pre-collected data and subsequently processed to determine parameters of the data where the distribution model is already known.

Two types of techniques are mentioned for non-parametric methods in [6] and are based on histograms and kernel functions. Histogram-based techniques represent the simplest technique of non-parametric methods, suitable for intrusion detection or fraud detection. For kernel function-based techniques, these functions can be used to estimate the probability distribution function for

standard data instances, and new instances are considered anomalous if they are in the low probability of this function.

3.2.4 Semi-parametric methods

Semi-parametric methods, according to [4], use several local distribution models instead of one global model. It is a combination of parametric and non-parametric methods, using the already mentioned core-based methods that calculate the density. Data that lie in the low-density region is considered anomalous. The more detailed characteristics of the algorithms for these methods are described in [4].

The authors in [6] state as an advantage of statistical methods the ability to work without a predefined labeled training data set. Another advantage may be the use of scoring, where the anomaly score is associated with a confidence interval, which may be helpful in making an overall decision about a given data instance. As a disadvantage, the authors point out the difficulty to select the right statistics or histogram-based techniques, where the algorithm searches for the anomalies, based on the values of individual attributes, however it does not evaluate combinations of attributes of a given instance.

4 Non-standard situation detection: The requirements

It is essential to realize the requirements of the proposed solution:

1. The algorithm should be able to learn the daily behavior of the user with respect to water consumption. It should be able to differentiate consumption in terms of minute, hour, day of week, and month of year. It is possible that the user has a higher daily consumption over the weekend than during the week. Also, in summer, the water consumption may be significantly higher than at other times. Consideration should also be given to whether it is hot or cold water.
2. The algorithm should not require the user's intervention during its operation but should be able to permit the user to evaluate the non-standard situations evaluated as standard. These adjustments should also affect the algorithm itself for future evaluation.
3. The algorithm should give a small number of false non-standard situations detected.

4. The user should be able to delete all their measured data and start learning again. This may be useful if the user changes and the same device is used again.

Meeting all above mentioned requirements should be enough to implement the solution that provides accurate and effective real-time detection of non-standard situations in the current water consumption with the minimum required human intervention, yet will also provide the feedback to intervene in decision process – whether the tested value is or is not anomalous.

5 Non-standard situation detection: The solution

The implemented experimental solution consists of three main parts: selecting an approach for anomaly detection, design of the algorithm for anomaly detection in water consumption and the implementation of the designed algorithm.

5.1 Selecting an approach for anomaly detection

Based on the analysis of non-standard situations and methods for their evaluation, described in previous parts, we distinguish three basic types of machine learning: supervised, unsupervised, and semi-supervised. All three types satisfy both the first and the fourth requirement, since they support continuous learning and the set of training data can still be erased. Supervised learning methods are highly accurate, satisfying the third requirement, but require a large set of training data, where each record has to be marked. To do this, a great effort is needed to correctly label data, which is contrary to the second requirement.

On the other hand, methods based on unsupervised learning meet this second requirement. Examples include clustering methods described in Chapter 3. However, the way clusters are formed is determined by an algorithm based on the search for cluster centers. These clustering centers are determined by the algorithm itself, given the number of points around the center. Figure 1 represents an example of a k-means algorithm with 12 clusters for water consumption data per hour on day of month-old data. In Chapter 3 one of three situations for anomaly detection is mentioned, where anomalous data is in small or sparse clusters. Applying this method to a twelve-cluster graph is represented by yellow and both light green clusters that

show non-standard values. The same is true for the blue cluster at the top right of the graph. These clusters are somewhat more accurate for sorting of non-standard values, however still contain normal water consumption values and thus continue to produce inaccurate detection results, which does not meet the third requirement.

Figure 1: Graphic representation of K-means algorithm for the current water consumption with respect to hours in day. Points are sorted into twelve clusters. Cluster centers are marked in gray.

Methods based on semi-supervised learning use the features of supervised and unsupervised learning. They can be applied to include a training set of only standard records, which means that during initial learning, the training set needs to be met by values from normal household behavior. At this stage, the necessary effort from the user can be expected to manipulate these values in the case of non-standard situations in the learning phase. However, at the end of this phase, the algorithm will already contain a minimum set of training data to predict future water consumption values based on the same or similar properties. Then, a number of user interventions in the algorithm's learning process is minimized. That means the second requirement can be considered fulfilled. The third requirement is met, given the increasing set of training data, the accuracy of the prediction to the next measured (test) value increases.

Based on these three approaches, it is best to use the features of semi-supervised learning to meet all the requirements in this paper.

Figure 2: Algorithm for detection of non-standard situation in water consumption.

5.2 Design of the algorithm for anomaly detection

The first step is to obtain a set of training data (see Figure 2). These training data will be stored in a remote database to which the algorithm will have direct access. Consequently, it is necessary to teach a model that will represent this semi-supervised learning. It is very important that the model learns to predict values based on only those features that are relevant to the outcome, otherwise the prediction can produce distorted results. It is also necessary to have the data at regular intervals, even in those where the consumption is zero, so that the model can correctly predict zero consumption. Then, the final consumption value can be defined by the month, day, hour, minute, and type features.

Once the model is trained, it is ready to estimate or to predict the next values. Only a single value representing the currently measured water consumption will be tested. The test value must be in the same form as the training data set, i.e., it must contain the same features. The resulting consumption value is not needed in this step.

If the model has predicted the future value based on the test features, the decision process follows. In this

step, the predicted value is compared with the actual measured consumption. Initially, the measured value must be greater than the predicted value. Lower water consumption than expected is not seen as a negative phenomenon within households for obvious reasons. Thus, if the value is higher than the predicted value, then the relative difference must be higher than the currently allowed one for the given measured value for a non-standard situation evaluation. According to authors Mateusz Mucha and Álvaro Díe [9], the relative difference between two values (a and b) is calculated as follows (see also Figure 2 – (1) and (2)):

$$R_d = \frac{|a - b|}{\frac{a+b}{2}} \quad (1)$$

The allowed difference D , expressed in percentages, is dependent on the predicted value P and is defined as follows:

$$D = \begin{cases} (80 - 10P)\% & (80 - 10P)\% \geq 20\% \\ 20\% & (80 - 10P)\% < 20\% \end{cases} \quad (2)$$

As this equation implies, the maximum allowed difference is 80%. The allowed relative difference decreases linearly with increasing predicted value, while the minimum relative difference does not fall below 20%. This relationship improves the efficiency of detecting non-standard values,

as at lower consumption values the relative difference is greater than at higher values. If the value is predicted at eight liters for a given situation, 10 liters is still acceptable, but in the case of a predicted one-liter consumption, the three-liter consumption is already too high.

The situations revealed by this equation as non-standard are stored in the database. Obviously, the training set needs to be expanded to make the other predictions more accurate. Since the goal is to leave only the standard data in this set, in this case it is not possible to save the actual water consumption. Therefore, the predicted water consumption is saved instead of the actual one. If the measured value was not marked as non-standard (it was lower than the predicted or the relative difference did not exceed the allowed value) the actual water consumption is stored in the training set. Subsequently, this extended set is stored in the database.

The designed algorithm would finish in the stage of extending the training set to a new record, ready to test and evaluate other records. Since the training set is stored in a database, it allows other software solutions to edit in this set and thus influence the learning itself. This is valid in cases when the user wants to change situations marked as non-standard or normal.

5.3 Implementation of the algorithm for anomaly detection

The algorithm is implemented in Python along with *scikit-learn* and *XGBoost* packages. Both of these packages contain libraries that are needed to implement machine learning in a semi-supervised manner. Three collections for MongoDB database were created:

- *measurements* – records of measured water consumption,
- *model_data* – learned data for model,
- *anomaly_data* – records of non-standard situations,
- *user_config* – user-specific configuration for learning mode.

Records that require testing against the non-standard situations are found by searching for all records in measurements that are not in *model_data*. The content of the entire *model_data* collection is then loaded. Since the date and time of the record is stored in a timestamp format (a non-negative natural number represented in seconds from January 1, 1970), it is necessary to extract this value to the mentioned features: *month*, *day*, *hour* and *minute*. Also, the models that allow value prediction need *DataFrame* input data that will consist of the following columns:

Month | Day | Hour | Minute | Type | Consumption

Subsequently, two frames are created from the data frame for the model - the first will contain the first four columns that will represent the features, and the second will contain the *Consumption* column, which is defined by these features. Both frames are used as input arguments for the *XGBRegressor* model from the *XGBoost* package to its saturation. After this step, the model attempts to predict actual consumption values based on new features that have been extracted from the frame for testing. The result is an array of predicted values in the decimal form, which are rounded by mathematical rules, since the actual water consumption in the system is given in whole numbers. By utilizing this field, the algorithm applies (2), where it determines whether or not a given record represents a non-standard situation. If true, an object is stored in the array of non-standard situations to represent the data structure of the measured record. A similar object is stored in another array of model data except that a rounded value of predicted consumption is used instead of current consumption.

6 Testing the prediction and detection

To prove that the experimental solution meets the required criteria, it needs to be tested. The testing of the algorithm consisted of two parts. As first, the actual consumption prediction was tested. Then, followed the detection of non-standard situations.

6.1 Testing the prediction of actual water consumption

Prediction of values is part of the solution for the evaluation of non-standard situations. The overall output of this algorithm is tested, *i.e.*, comparing predicted values to actual ones. Two test cases were included for model data:

1. Testing of data prediction with model data per one week.
2. Testing of data prediction with model data per one month.

Both cases were tested for the seven days awaiting evaluation. Tables 1 and 2 show the results for the 22nd hour on the second test day to find significant differences. It can be seen that the predictions for monthly learning are more accurate when compared to weekly learning. It is also sup-

Table 1: Sample data output from testing of water consumption prediction values with model data per week [in liters].

Date	Time	Water type	Predicted	Actual
02.03.	22:03	Cold	0	1
02.03.	22:04	Hot	0	0
02.03.	22:19	Cold	0	1
02.03.	22:20	Hot	0	0
02.03.	22:31	Cold	0	1
02.03.	22:32	Hot	0	1
02.03.	22:47	Cold	1	1
02.03.	22:48	Hot	0	0

Table 2: Sample data output from testing of water consumption prediction values with model data per month [in liters].

Date	Time	Water type	Predicted	Actual
02.03.	22:03	Cold	1	1
02.03.	22:04	Hot	0	0
02.03.	22:19	Cold	1	1
02.03.	22:20	Hot	0	0
02.03.	22:31	Cold	1	1
02.03.	22:32	Hot	0	1
02.03.	22:47	Cold	1	1
02.03.	22:48	Hot	0	0

ported by the fact that the average difference between the predicted value and the actual value is 0.33 liters after the monthly learning and 0.4 liters after the weekly learning on a given sample. The percentage success rate that the predicted value was the same as the actual value is 64% after weekly learning, and 70% after monthly learning. This comparison demonstrates that with increasing learning time, the accuracy of the prediction of measured values improves.

6.2 Testing the non-standard situation evaluation

The marking of the currently measured value as non-standard depends on the prediction of the values for the given parameters or features. As in the previous testing, both model data and weekly and monthly data was tested here as well.

Sixteen non-standard situations arose in the test data for the first week of March. Tables 3 and 4 show a list of the first ten detected non-standard situations. These include the day and time of the non-standard situation, the type of

Table 3: Output of the first ten output records of testing the evaluation of non-standard water consumption situations with model data per week [in liters].

Date	Time	Water type	Predicted	Actual	Non-stand.
01.03.	12:20	Hot	0	2	No
01.03.	20:16	Cold	1	3	Yes
01.03.	23:33	Hot	0	2	No
02.03.	10:33	Hot	0	2	No
02.03.	11:02	Cold	1	3	Yes
02.03.	13:02	Hot	0	3	Yes
02.03.	20:20	Hot	0	2	No
02.03.	20:34	Hot	0	3	Yes
02.03.	10:17	Cold	1	3	Yes
02.03.	20:20	Hot	0	2	No

Table 4: Output of the first ten output records of testing the evaluation of non-standard water consumption situations with model data per month [in liters].

Date	Time	Water type	Predicted	Actual	Non-stand.
01.03.19	12:03	Hot	0	2	No
01.03.19	12:20	Hot	0	2	No
01.03.19	12:33	Hot	0	2	No
01.03.19	20:16	Cold	1	3	Yes
01.03.19	22:48	Hot	0	2	Yes
01.03.19	23:33	Hot	0	2	No
02.03.19	10:33	Hot	0	2	No
02.03.19	11:02	Cold	0	3	Yes
02.03.19	13:02	Hot	0	3	Yes
02.03.19	20:47	Hot	0	2	No

water, the predicted and actual consumption, and whether or not it was an actual non-standard situation.

By comparing these tables, it can be seen that the evaluation of non-standard situations for model data per one month detected more false positives than model data per one week. However, the total number of non-standard situations found was 23 after the monthly learning and 29 after one-week learning, while all the actual non-standard situations found in the weekly learning were also found in monthly learning. In addition, Friday's record at 22:48 (Table 4) was not detected as a non-standard situation for one-week model data, however, was detected on model data for one month, which was evaluated correctly. Out of a total of 16 actual non-standard situations, 14 and 15 were identified at weekly and monthly learning, respectively. In both cases there was a false negative evaluation of the records,

yet following the results of this test it can be said that longer-term learning produces more accurate results, with fewer false positives and false negatives.

The output of all tests showed that the individual algorithms implemented in the system work properly. In case of one-month learning, the estimated values were more accurate when compared to one-week learning. Also, the overall average difference between the estimated value and the actual value was lower after a month of learning. The algorithm for estimation of non-standard situations based on the prediction contained a smaller number of false positives after one month of learning when compared to one week of learning. Thus, it can be assumed that longer-term learning will continue to produce significantly more accurate results and the number of false-positive non-standard situations will decrease.

7 Experimental hardware solution

The hardware solution for water metering was already implemented and experimentally verified [10]. However, due to the nature of non-standard situation detection the processing of the data could not be handled by the hardware. Thus, it was decided to place it on the server and install an additional camera (for monitoring both types of water) and deploy some form of light, activated only when necessary.

7.1 From the collector to the server

Data is acquired using the experimental device located close to the water meter. The first step is to implement the algorithm for reading the digits and for evaluation of the current consumption on the server. Note that previously this was implemented as a part of the experimental device. Thus, the algorithm will reside on a server and will run for each pending image. Its execution is handled by the Shell script that goes through the folder of images to be processed; once processed, they are moved to a subdirectory. All processed images are already in this subdirectory.

The experimental device with the camera should only read water meter values (*i.e.*, provide image).

It is difficult to store each image of the water meter on the server due to the limited capacity of memory. Such an approach is quite inefficient, especially if the values on the meter do not change. For this reason, it is a good idea to minimize the amount of stored images only in case when change occurs. Applying a simple but effective solution to detect changes in the image would greatly reduce the num-

ber of images. However, the design of the algorithm for detection of non-standard situations requires data storage at regular intervals, even if no shift of the dials on the water meter was detected. Thus, it still is valid to send requests to the server. These requirements can then be divided into two types:

- send the image once the change occurs,
- send request without image, no change occurs.

In the proposed algorithm, the first step is to log on to the system. Once logged in, each image is then requested at fifteen-minute intervals, while comparing the changes to the previous frame. If the change is sufficient to conclude whether the actual change of digits occurred, a request is sent to the server for the current image of the meter. The image is required for the calculation of the new current consumption. Otherwise, a request will be sent to the server without the image and the server will be only informed of the zero consumption. The term change in the image means the shift of the dials on the water metering device.

The implementation of the proposed algorithm in Python solves the change in the image at the pixel level. The principle of the algorithm is that it passes the new image pixel by pixel and compares it with the previous one. In comparison, it focuses only on the green channel, as it is the highest quality channel. However, a new image with dimensions of 160×120 pixels is captured first. Then, the absolute value of the pixel difference is compared with the previous stored image in the memory, and if this value is higher than the threshold, it increments the number of changes by 1. If this number of changes is larger than the allowable value, the image is considered as changed.

As for logging into the system, this is accomplished by creating an authentication request to the remote server that, in the case of successfully entered credentials, provides the device with an authentication key, a so-called token that will be attached to each authentication request. Three attempts to send an image are made, if the previous was unsuccessful. However, prior to that, it will attempt to request a new authentication key if the previous one is no longer valid. If successful, the server saves the image in a folder for further processing. Images that are ready to be sent to a remote server are also stored in the experimental device. To avoid depletion of the storage space, a check is performed prior to saving to ensure enough memory is available. If not, the algorithm will delete old images until enough memory is freed.

7.2 Additional camera

Another step is the installation of an additional camera to capture the second water meter (hot or cold water). Because the current experimental device allows us to install only one camera with a Camera Serial Interface (CSI), a USB webcam was used. Optimal is to have camera as small as possible, since the space in which the camera is to be located is confined.

7.3 Additional light

Another thing to bear in mind when proposing changes to the current hardware solution is the quality of light. Thus, proposed prototype improvements are to activate the light only when the camera takes a picture. In the original hardware solution [10], the light source was under constant current. Instead, the so-called General-Purpose Input / Output (GPIO) pins can be used, which are programmable and thus can be switched on or off. The maximum voltage of the GPIO output pins is 3.3V. It is thus necessary to find an illumination capable of emitting sufficiently strong light at this voltage. Two LED diodes were used together, one for each camera.

Considering the above, the algorithm for capturing of the image of the water meter was extended as follows:

1. Activate light via GPIO.
2. Capture the image of the water meter.
3. Deactivate light via GPIO.

7.4 Putting it all together

The implementation of the proposed prototype extension is illustrated in Figure 3. LED diodes are attached to the cameras and activated when the image is captured and deactivated once the image is taken. The GPIO pins by default produce a low voltage. As soon as the operating system is loaded, the device is set to run three programs in the following order:

1. Run a program to deactivate both LED diodes.
2. Run the program for capturing the first water meter using CSI camera.
3. Run the program for capturing the second water meter using USB camera.

In addition to the first program, the other two run in the background as they are not end state. Also, the time between the start of the second and third program is one minute to avoid intermittent illumination.

Figure 3: Implementation of the experimental prototype hardware.

8 Conclusion

In this paper, the experimental solution for detection of a non-standard situation in current water consumption based on machine learning is described. Thanks to the semi-supervised approach, the described solution offers a quite accurate prediction for future water consumption. Based on the predicted value, an evaluation is made to determine whether or not a value represents the non-standard situation with respect to hour in day, day in week and the current month. The evaluation process uses the percentage difference between the predicated value and the actual one. Also, the value of the allowed percentage difference for a given predicted value is estimated. If the calculated percentage difference is greater than allowed, the test value is evaluated as non-standard.

The algorithm for detecting the change in the image was also designed and implemented. Thus, it is possible to send an image of the water meter to the server once the dials move. In addition, an additional camera and two LED lights were installed to illuminate the space when the cameras are activated.

This proposal was tested on two scenarios – with learned model data for a week and with learned model data for a month. Test results have shown that longer-term learning provides a more accurate prediction and fewer false-positive and false-negative findings of non-standard situations. Accordingly, the described solution is sufficiently effective and accurate for a real-time detection of non-standard situations in the current water consumption.

References

- [1] Penttilä J., A method for anomaly detection in hyperspectral images, using deep convolutional autoencoders, 2017.
- [2] Dorj E. and Altangerel E., Anomaly detection approach using hiddenmarkov model, in *Ifoist*, 2013, 2, 141–144.
- [3] Carvalho L., Teixeira C., Dias E. C., Meira W., and Carvalho O., A simple and effective method for anomaly detection in health-care, *Proceedings of the SIAM International Conference on Data Mining Workshop*, 2015, 16–24.
- [4] Hodge V. and Austin J., A survey of outlier detection methodologies, *Artificial intelligence review*, 2004, 22(2), 85–126.
- [5] Geijer C. and Andreasson J., Log-based anomaly detection for system surveillance, Ph.D. dissertation, Masters thesis, Chalmers University of Technology, Gothenburg, Sweden, 2015.
- [6] Chandola V., Banerjee A., and Kumar V., Anomaly detection: A survey, *ACM computing surveys (CSUR)*, 2009, 41(3), 15.
- [7] Friedman J., Hastie T., and Tibshirani R., *The elements of statistical learning*. Springer series in statistics New York, 2001, 1(10).
- [8] Amer M. and Goldstein M., Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer, *Proceedings of the 3rd Rapid-Miner Community Meeting and Conference*, 2012, 1–12.
- [9] Mucha M. and Díez Á., Percentage difference calculator. [Online]. Available: <https://omnicalculator.com/math/percentage-difference>
- [10] Petija R., Kainz O., Dujava M., Alexandrova G., Fecilak P. and Moravcik M., Measurement of Water Consumption based on Image Processing, in *International Conference on Emerging eLearning Technologies and Applications*, 2019.