2020 Global Smart Industry Conference (GloSIC)

Cleaning Sensor Data in Smart Heating Control System

Mikhail Zymbler
Computer Science Department
South Ural State University (national research university)
Chelyabinsk, Russian Federation
mzym@susu.ru

Yana Kraeva
Computer Science Department
South Ural State University (national research university)
Chelyabinsk, Russian Federation
kraevaya@susu.ru

Elizaveta Latypova
Computer Science Department
South Ural State University (national research university)
Chelyabinsk, Russian Federation
latypovaea@susu.ru

Sachin Kumar
Computer Science Department
South Ural State University (national research university)
Chelyabinsk, Russian Federation
sachinagnihotri16@gmail.com

Dmitry Shnayder
Automation and Control Department
South Ural State University (national research university)
Chelyabinsk, Russian Federation
shnaiderda@susu.ru

Alexander Basalaev
Automation and Control Department
South Ural State University (national research university)
Chelyabinsk, Russian Federation
basalaevaa@susu.ru

***Abstract*—Sometimes, smart heating control applications are partially equipped with missing values and outliers in the sensor data due to software/hardware failures/human errors. To provide an effective analysis and decision-making, erroneous sensor data should be cleaned by imputation of missing values and smoothing outliers. In this paper, we present a case of the Smart Heating Control System (SHCS) installed in the South Ural State University, and describe the structure and development principles of Data Cleaning Module (DCM) of the system. We implement DCM through data mining and neural network technologies as a set of the following subsystems. The preprocessor extracts raw data from the system's data warehouse and prepares a training data for further processing. Predictor provides Recurrent Neural Network (RNN) to forecast the next value of a sensor based on its historical data. Reconstructor determines if the current value of a sensor is an outlier, and if so, imputes it by the synthetic value from Predictor. Finally, Anomaly Detector subsystem discovers anomalous sequences in the sensor data. In the experiments on the real sensor data, DCM showed relatively high and stable accuracy as well as adequate detection of anomalies.**

***Keywords*—sensor data, smart heating control, time series, missing data imputation, outlier detection, anomaly detection, recurrent neural network**

I. INTRODUCTION

Modern day Smart Heating Control Systems (SHCS) employ various sources of data, including the data from meter's measurements related to utilities consumption, data from process controllers and indoor climate control sensors. Implementation of the IoT technology in systems of this kind additionally enables receiving some sizeable sets of data on various parameters having an impact on the overall heating conditions in a building. The resulting generation of big data allows one to conduct comprehensive analysis of heating systems, enabling timely identification of instances of outlying performance values and out-of-range energy efficiency indicators of dwellings.

However, in keeping with the reliability theory – when reviewing the entire population of data within one single system, the growth of the number of data sources may lead to data analysis inaccurate, in an unfortunate situation where even one data source would fail.

Occurrence of invalid data, or outliers, may be due to a variety of factors: equipment wear or faulty installation or operation (generally referred to as the human error factors), software and hardware design flaws, operating conditions being outside the permissible or design range, etc.

Invalid data readings or input may lead to a decrease in the energy efficiency of dwellings over the course of their use, resulting in fallacious control exercised in automatic and automated modes, and at times leading to energy penalties from the power suppliers. With all this in mind, the task becomes current and topical for us to come up with a viable solution for prompt data cleaning, which should comprise timely identification of outliers and invalid data readings or inputs, as well as online reconstruction of the missing data.

The South Ural State University (SUSU) is an academic and research institution possessing advanced competences in the field of automated heat supply control systems. The university develops and implements automation solutions for its utility networks such as heat, water, gas and power supply systems. To that end, the SUSU campus has received a new Smart Heating Control System based on the SCADA system named PolyTER [5], which allows monitoring and control of operating conditions of utility systems of the university campus buildings, comprising both wired and wireless IoT sensors. Based on the data obtained on the energy consumption of various facilities and dwellings, experts analyze their energy-related performance.

The PolyTER SCADA-system, much like other similar systems, has its basic capabilities of identifying outliers in data sets, and generating related outlier notification alarms specific to the individual subsystems. However, after identification, it often takes overly long to address and resolve certain problems

The study was financially supported by the Ministry of Science and Higher Education of the Russian Federation within the framework of the Russian Federal Program for the Development of Russian Science and Technology from 2014 to 2020; project identifier: RFMEFI57818X0265 (contract no. 075-15-2019-1339 (14.578.21.0265)).

978-1-7281-8075-5/20/$31.00 ©2020 IEEE

that may have led to the actual outliers. Moreover, to be unambiguously identified, some faults require a deeper analysis. In particular, such tasks include identifying areas of the system, whose heat supply faults will lead to the unbalancing of generation and consumption of thermal resources, which may negatively affect the process of comprehensive integrated optimization of the heating system, normally performed in real time. Because of this, the application of methods for cleaning and reconstruction of data is an important basic task that needs to be addressed in this system.

In this paper, we presented an approach to cleaning sensor data in the Smart Heating Control System of SUSU. The rest of the paper is structured as follows: In section II, we give overview of the Smart Heating Control System. Section III briefly discusses related work. The structure and development principles of Data Cleaning Module are described in Section IV. Results of the experimental evaluation of our approach are presented in Section V. Section VI concludes the article with a discussion on future scope.

II. Overview of SUSU Heating Control Systems

The Smart Heating Control System of SUSU is a cyber-physical system that encompasses various server and network equipment items, metering devices for energy consumption and generation monitoring, process controllers for the heating fluid and hot water supply, as well as more than 1000 various sensors.

Fig. 1 shows the structure of the SUSU Smart Heating Control System. At the lower level, the System includes various wired and wireless sensors, metering devices and controllers. At the intermediate level, the communication of controllers and the metering units with the database server is enabled and maintained by means of various wired and wireless network equipment. The third level includes a SCADA system with a database server used for processing of information. The resulting processed data is then transmitted to the workstations of local net users via the SUSU LAN or to the remote users' workstations over the Internet.

The first steps to implement parts of this system at SUSU were taken in 2010. In introducing this system at SUSU, special attention was paid to the heating domain. The objective of this implementation then was the overall process control optimization of heat supply and heat consumption through integration of geographically distributed measuring instruments into the system of automatic control of the University's own district heating co-generation and distribution systems, both on a centralized basis and locally in individual heat energy consumers.

In 2018, the sensor subsystem was significantly expanded with new IoT devices (e.g. over 300 wireless temperature sensors were then installed), which permitted the SUSU in acquiring additional information on the temperature conditions of the dwellings, making it possible to significantly optimize heating of the buildings. Implementation of Smart Heating Control System at the SUSU campus allowed SUSU to mark the saving of approximately 15% of the heat energy as compared to the historical consumption data.

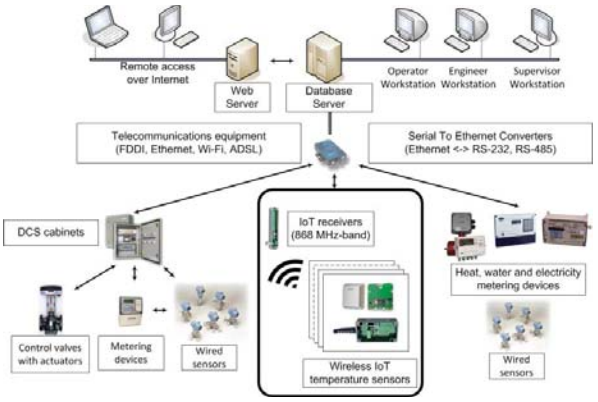


Fig. 1. Architecture of SUSU automated utilities control system

A. Software and Hardware Components

The system consists of the following measuring and control equipment:

- heat energy metering units (57 pcs.) equipped with a heat calculator integrated with 2 pressure sensors, 2 temperature sensors and 2 flow meters;
- cold water metering units (58 pcs.), equipped with a calculator featuring integration with a pressure sensor and a flow meter;
- gas metering units (3 pcs.) equipped with a gas volume corrector featuring integration with temperature and pressure sensors and a flow meter;
- process controllers to control distribution of the heat energy at individual district heating substations (24 pcs.) with 5 temperature sensors connected to them and various discrete (Y/N) protection sensors or switch-type probes, as well as the actuators for the control and shut-off valves.
- process controllers for the control of generation and distribution of heat and electrical energy (5 pcs.) of the CHP plants, of the gas boiler house and central heating units with a multitude of various sensors and control instruments and actuation devices.
- wireless sensors for indoor temperature monitoring (a total of over 300 pcs.).

The core of the system is the PolyTER software package based on the C++ programming language. This software supports open and proprietary communication protocols with equipment from various manufacturers. It has a configurable data visualization environment, and SMS and e-mail notification capabilities. Oracle Database is used as a database management system. The R programming language is used for data analytics.

B. The Need and Basic Functions of Data Cleaning

The main problem in the operation of measuring equipment is the recurrent data output errors or outlying data due to perturbation inputs of indeterminate or arbitrary nature.

One of the possible relevant causes is equipment failure. About 10 to 20 devices have to be replaced annually over equipment obsolescence or inadvertent departure from the intended operating conditions or haphazard. Another reason is the loss of the necessary contact with the measured heat medium, e.g. due to pressure gauge clogging etc. Besides, interruption of communication links with field instruments and sensors may be the recurrent factor having its fair share of contribution. Moreover, not all instruments feature internal hardware-based data archiving capability, which is especially true for controllers. Another cause may be the faulty installation of equipment, which may lead to extra perturbation inputs associated with abnormal operating conditions of sensors in the actual areas of their faulty placement.

Occurrence of invalid or outlying data leads:

- to invalid erroneous calculations occurring when optimization algorithms are run, to build incorrect properties of the controlled objects or processes and consequently to wrong strategic decision making;
- to incorrect administrative managerial decisions taken by personnel when faced with the misleading and false deviations of the current performance indicators from their intended rated values and ranges;
- to fallacious automatic control exercised by the process controllers, which, in turn, may not only lead to excess energy penalties, but also to failure of district and local utility networks (e.g., freeze-ups of the heating system pipelines);
- to metering device sensor failures, in which case erroneous calculation of utilities consumption may occur, not to mention the possible energy penalty charged by the power provider to the power consumer for untimely detection and late or missing remedial action.

Fig. 2 shows an example of one of the problems associated with the failure of the power supply unit of a flow meter on the return pipeline, which led to the false negative heat load in an open circuit heating system.

In conditions of the voltage drops that occurred at the mains, the power supply unit failed to provide the necessary stable voltage level to properly power up the flow meter, thus triggering constant consecutive reboots of the flow meter firmware, which caused the instrument to transmit spurious, unwanted signals to the heat meter at each of these abnormal startups.

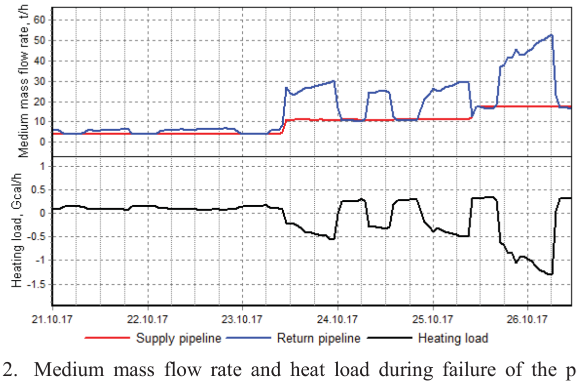


Fig. 2. Medium mass flow rate and heat load during failure of the power supply unit of a flow meter

Therefore, timely detection, and rectification of equipment malfunctions and data reconstruction have a significant administrative and economic effect by reducing or eliminating loss of benefit, when such equipment problems occur.

The data cleaning module operation accomplishes the following main purposes:

1) online detection of gaps and outliers in the sensor measurement data and replacement or filling of such outlying values or empty value slots with plausible deductive synthetic values;

2) online detection of anomalous sensor behavior and notifying the operator of the anomalies discovered.

The data cleaning module is integrated into the system between the level of data reading and the level of use of the data for analytical calculations, and visualization or representation. In the meantime, the visualized data are marked as either original or reconstructed, with the access conveniently retained to the outlying data for possible deeper analysis of the same, in case of need.

III. Related Work

Several techniques are devoted to outlier detection; many of them are used in fault detection theory [19], [22].

The basic approaches include the methods that define and thus help to identify departures of actual values beyond the permissible deviation spread based on the values obtained by object simulation [18], [24].

In research paper [20] for the ventilation and air conditioning monitoring systems, the authors employed a combination of three techniques for online fault detection, namely reduction of dimensionality of correlated data which indicate occurrence of outliers (ReliefF method + Adoptive Genetic Algorithm); application of Extended Kalman Filter for noise filtering and data decomposition into time series; separation of areas of valid and invalid values of static and dynamic parameters of the object model using recursive one-class SVM.

To determine deviations in the district heating substation (DHS) operation in research paper [9], the authors proposed the reference-group (peer-group) based approach to operational monitoring. The performance parameters of a group of similar DHSs are considered as a reference, based on the k-most similar objects criterion using the Euclidean distance. Deviations of the performance of one DHS from its kin group's indicators by the value of the relevant threshold are regarded as outliers, thus enabling possible fault detection at the DHS.

Research paper [11] exploits the Principal Component Analysis (PCA) method for identification of outliers in the performance data of HVAC-systems whereas the authors of research paper [8] described the data reconstruction method used in conjunction with the PCA.

The balance models construction technique is also noteworthy. The heat power or hydraulic imbalance in such models indicates presence of faults in the system or presence of outliers [13].

Reconstruction of data lost to the outliers is performed by predictive techniques. A lot of research work was devoted to predict the parameters of heating systems using static and dynamic factor models and time series models with relevant identification enabled by the autoregressive analysis [23], support vector machine (SVM) [12], and recurrent neural networks [2].

Our interest was also attracted by an approach to reconstruct the static and dynamic properties of models, whose distortion occurs because of variable thermal perturbation inputs. Such perturbation inputs are considered by virtue of their description through schedule based indicator functions [6].

In the meantime, it is noteworthy that the peculiarity of the performance data of such heat energy supply systems is the periodic or recurrent nature associated with the cyclical change in weather conditions and the intended operating conditions of dwellings in function to the time of the day. The performance data outliers that occur in the course of operation of the various heating subsystems are characterized by abrupt changes in the signal values, with these surges lasting for limited periods.

Based this, our research proposes the use of long short-term memory recurrent networks to detect outlier sequences, and to reconstruct the lost sequences of performance measurement data, which have a cyclical changing nature to them.

IV. Development of the Data Cleaning Module

In this section, we describe module structure and development principles of *DCM*.

A. General Structure

Fig. 3 depicts overall workflow of *DCM*. *DCM* is developed for each single sensor of the cyber-physical system, and consists of four subsystems, namely *Preprocessor*, *Predictor*, *Reconstructor*, and *Anomaly Detector*. The *Preprocessor* subsystem prepares sensor data for further processing. The *Predictor* subsystem provides an artificial neural network (ANN) to forecast the next value of a sensor based on its historical data. The *Reconstructor* subsystem determines if a given value of a sensor is an outlier, and if so, imputes it by the synthetic value received from *Predictor*. Finally, the *Anomaly Detector* subsystem discovers anomalous sequences in the sensor data.

Workflow of *DCM* for a specified sensor explained as follows. *Preprocessor* performs its actions regularly with the δ_1 period of time. Such a period is predefined by an operator of the cyber-physical system with the typical value δ_2=3 months. *Preprocessor* extracts a part of the sensor data accumulated up to the current time, prepares a training set for *Predictor*'s ANN.

Then, the following actions are performed regularly with the δ_2 period of time where $\delta_1 \gg \delta_2$. Such a period is also predefined by the operator of the system, and its typical value is δ_2=300 seconds. ANN predicts the current value of the sensor. If the real value returned by the sensor is NULL (i.e. it is missed) then current value is changed to the value predicted by ANN. If the real value is not missed, then *Reconstructor* checks if it is an outlier or normal. If the real value is recognized as "normal", then *DCM* passes it to the system to save in the data warehouse. Otherwise, *Reconstructor* changes the current value to the value produced by *Predictor*. Finally, *Anomaly Detector* determines if the current value ends some anomalous sequence of sensor values, and if so, notifies the operator.

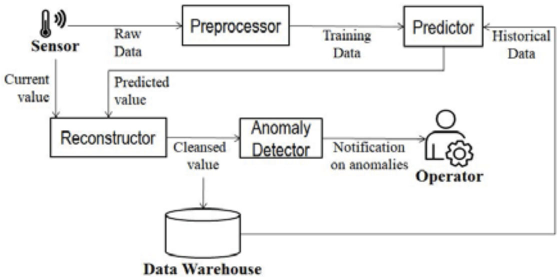


Fig. 3. Overall workflow of DCM.

Below, we give more detailed description of the above-mentioned subsystems.

B. Preprocessor

Preprocessor is aimed to prepare a training set for the learning ANN of the *Predictor* subsystem. *Preprocessor* consists of the following subsystems, namely *Parser*, *Restorer*, *Outlier Detector*, and *Normalizer*.

Preprocessor performs as depicted in Fig. 4. At first, *Parser* extracts a part of the sensor data from the data warehouse, and transforms it in an appropriate way to be processed further. Next, *Restorer* imputes missing values in the parsed sensor data. Then, *Outlier Detector* finds points in the data that deviate significantly from the rest data points. All the outliers found are substituted by NULLs as if they missed, and *Restorer* performs imputation once again. Finally, *Normalizer* splits imputed sensor data into a set of fragments, and performs normalization of each fragment. These steps result in a data file to learn ANN of the *Predictor* subsystem.

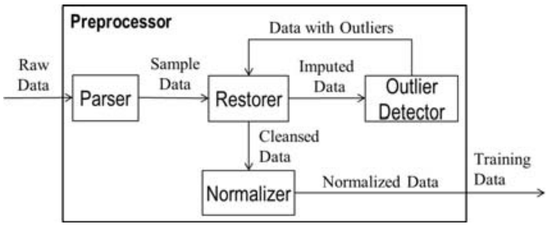


Fig. 4. Workflow of Preprocessor.

Normally speaking, *Parser* extracts a *time series* T, which is a chronologically ordered sequence of real-valued elements (with a small fraction of empty values): $T = (t_1, \dots, t_m)$ where m called length of time series, and $t_i \in \mathbb{R}$ or $t_i =$ NULL. Then, *Restorer* and *Outlier Detector* together transform T so as $\forall i\ t_i \in \mathbb{R}$.

Normalizer produces S_T^n, a set of normalized subsequences of T with length n $(n \ll m)$, and P, a set of respective predictions. A subsequence of time series T is its contiguous subset consisting of n elements, and starting from the given position i : $T_{i,n} = (t_i, \dots, t_{i+n-1})$, $1 \le i \le m - n + 1$. A normalized subsequence is calculated by min-max scaling as follows: $\tilde{T}_{i,n} = (\tilde{t}_i, \dots, \tilde{t}_{i+n-1})$ where $\tilde{t}_i = \frac{t_i - t_{min}}{t_{max} - t_{min}}$, and $\tilde{t}_i \in$

$[0; 1]$. For a normalized subsequence $\tilde{T}_{i,n}$ an element $\tilde{t}_{n+1}$ is treated as its prediction.

The length of the subsequence is calculated as $n = f \cdot h$ where f is the sensor frequency, and h is time interval in the past used by *Predictor* (historical horizon). The latter is a parameter predefined by an operator of the cyber-physical system. For instance, if the sensor frequency is 4 times an hour and the historical horizon is 12 hours, then the length of sequence is 48.

Preprocessor is implemented by several Python libraries as follows. *Parser* uses the standard openpyxl and pandas libraries. *Outlier Detector* is based on the algorithms from the adtk (Anomaly Detection Toolkit) library [3]. *Restorer* exploits the ARIMA (AutoRegressive Integrated Moving Average) model [7] implemented in the standard statsmodels library. *Normalizer* is implemented by the standard sklearn library.

C. Predictor

Predictor provides a recurrent neural network (RNN) with long short-term memory (LSTM) [10] layer. We learn RNN on the set of normalized subsequences of the sensor data prepared by *Preprocessor*. When using, RNN takes a subsequence of the real sensor values preceding to the predicted value as an input, and outputs the predicted value.

RNN performs as depicted in Fig. 5. The LSTM layer is composed of LSTM cells (memory blocks) where the number of cells is equal to the subsequence length chosen at the preprocessing step. Performing together, LSTM cells produce column vector h , so-called hidden state. Length of this vector is a parameter predefined by an operator of the cyber-physical system. The Dropout layer randomly deactivates ζ percent of neurons in the h vector to prevent overfitting of RNN. The ratio of deactivated neurons is also a parameter of the system with typical value $\zeta = 20\%$. The Dense layer applies rectified linear unit (ReLU) as an activation function to transform data to the single predicted value $\hat{t}_{i+1}$.

Each LSTM cell consists of a cell state and several gate layers. The cell state is a vector that carries the information from the previous moments and will flow through the entire chain of LSTM cells. The LSTM cell has three gate layers, namely the input gate, the forget gate and the output gate, which regulate the amount of the data should be kept, forgot and delivered to the output, respectively.

Predictor is implemented by the Keras library [16] and the TensorFlow framework [1].

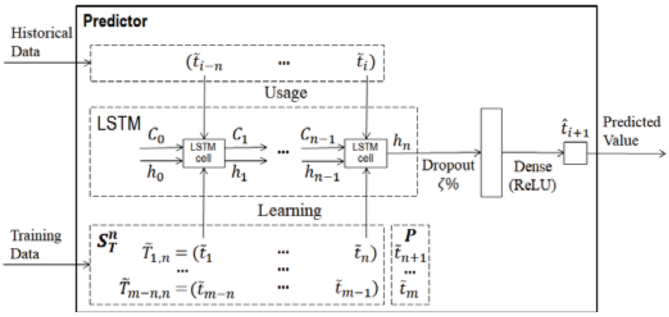


Fig. 5. Workflow of Predictor.

D. Reconstructor

Reconstructor takes the current non-NULL value of the sensor, and checks if it is significantly dissimilar to the value produced by *Predictor*. If so, the real value is recognized as an outlier and changed by the synthetic value from *Predictor*.

Implementation of *Reconstructor* is based on the probability distribution of the prediction error [17]. Following this method, we should determine dissimilarity threshold $\varepsilon > 0$ for a real and a synthetic value of the sensor, t_{i+1} and $\hat{t}_{i+1}$, respectively. If $|t_{i+1} - \hat{t}_{i+1}| \geq \varepsilon$, then t_i represents an outlier. The dissimilarity threshold is calculated as $\varepsilon = \mu + k\sigma$ where μ is mean of prediction errors, σ is standard deviation of prediction errors, and $k > 0$ is predefined parameter of the system (with typical value $k = 3$). A single prediction error is calculated as an absolute difference between the last point of a subsequence from the *Predictor*'s training set and corresponding synthetic value produced by *Predictor*.

E. Anomaly Detector

Anomaly Detector takes a set of subsequences in the time series of the sensor data ended by the current value and determines if each subsequence of the set is significantly dissimilar to the rest subsequences of the series, and if so, notifies the operator of the cyber-physical system.

The number of such subsequences is a parameter predefined by the operator so that each subsequence length corresponds to some typical time interval in the subject domain. Therefore, the subsequence length is calculated depending on the sensor frequency. For instance, if the sensor frequency is 4 times an hour and the operator would like to be notified on possible anomalies in the sensor data in the nearest 12 hours, 1 day, and 2 days, then *Anomaly Detector* will try to determine anomalous subsequences with lengths 48, 96, and 182 points, respectively.

Implementing *Anomaly Detector*, we exploit the discord concept [14], [21]. A discord looks attractive as an anomaly detector because it only requires one intuitive parameter (the subsequence length), as opposed to most anomaly detection algorithms, which typically require many parameters [15]. Discords are formally defined as follows.

Two subsequences $T_{i,n}$ and $T_{j,n}$ are *non-trivial matches*, if $\exists T_{p,n} \in S_T^n, i < p < j: \mathrm{ED}(T_{i,n}, T_{j,n}) < \mathrm{ED}(T_{i,n}, T_{p,n})$ where $\mathrm{ED}(\cdot,\cdot)$ denotes the Euclidean distance. Let M_C denotes a non-trivial match of a subsequence $C \in S_T^n$. Then a subsequence $D \in S_T^n$ is said to be the *most significant discord* in T if $\forall C \in S_T^n \min(\mathrm{ED}(D, M_D)) > \min(\mathrm{ED}(C, M_C))$, i.e. if the Euclidean distance to its nearest non-trivial match is the largest. A subsequence $D \in S_T^n$ is called the *most significant k-th discord* in T if the distance to its k-th nearest non-trivial match is the largest.

Thus, *Anomaly Detector* notifies the operator if a subsequence with the predefined length and the current sensor value at its end is the most significant k-th discord where k is also a parameter predefined by the operator of the system. Implementation of *Anomaly Detector* is based on the MatrixProfile library for Python [4].

V. Experimental Evaluation

We evaluated the proposed approach during the experiments conducted on the real sensor data taken from SUSU SHCS data warehouse.

We assessed the accuracy of *DCM* as follows. We took the 2018 year data of a sensor installed in a lecture hall, and pass it to *Preprocessor*. Then, we learned *Predictor*'s RNN by 80 percent part of resulting data produced by *Preprocessor*. Finally, we simulated every-day work of *DCM* on the block of rest 20 percent part of the *Preprocessor* resulting data, treating it as a test set.

In the experiments, we configured *DCM* subsystems follows. To learn *Predictor*'s RNN, we used subsequences of length n=48 corresponding 12 hours of the sensor work. In RNN, we took hidden state h as a column vector of length $|h|$=32. We used MSE (mean square error) as a loss function, Adam as an optimizer while learning, 15 epochs, and size of batch 32.

As accuracy measure, we used the *root mean square error (RMSE)*, defined as $\text{RMSE} = \sqrt{\frac{1}{|B|}\sum_{i=1}^{|B|}(t_i - \hat{t}_i)^2}$ where t_i and $\hat{t}_i$ are the real and synthetic sensor values, respectively, and $|B|$ denotes the block length. Fig. 6 shows the experimental results regarding RMSE where we took block length from one week to two months (i.e. up to full length of the series in the test set).

As can be seen, *DCM* provides relatively high and stable accuracy. Fig. 7 visualizes two excerpts from simulation of *DCM* work, namely for one-month and two-month block length. As can be seen, *DCM* adequately predicts normal values as well as detects outliers.

Fig. 8 depicts an example of two anomalies found during the simulation of *DCM* work. Both anomalies correspond to two days activity of the above-mentioned sensor. The first one is top-1 discord in the test set, and may indicate that the sensor was temporarily out of order. The second anomaly represents top-10 discord in the test set, and may indicate fast decrease of temperature in the lecture hall because of intensive ventilation due to the large number of open windows on a hot day. Anyway, anomalies detected are the subject of the operator's reaction.

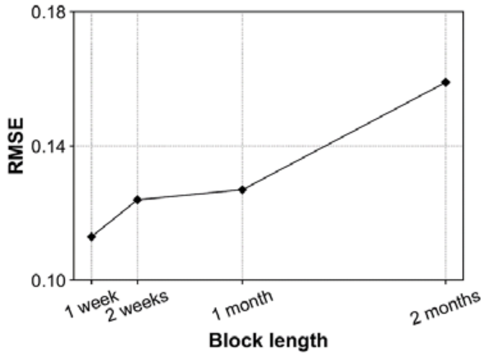
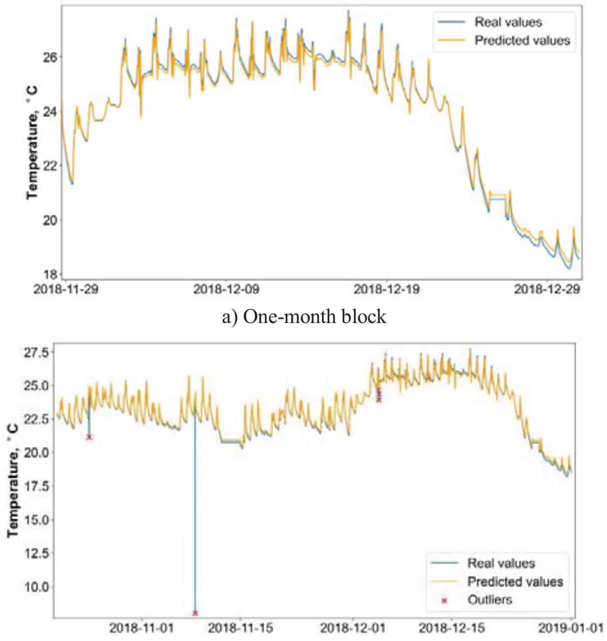


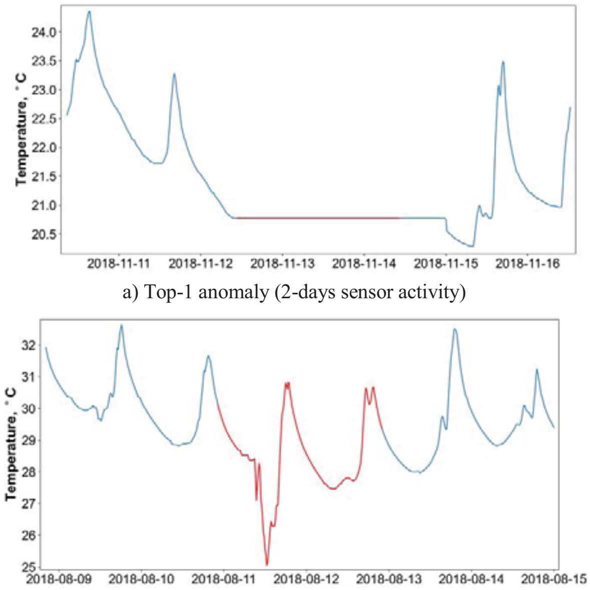
Fig. 6. Accuracy of DCM.



a) One-month block

b) Two-months block

Fig. 7. Simulation of DCM work.



a) Top-1 anomaly (2-days sensor activity)

a) Top-10 anomaly (2-days sensor activity)

Fig. 8. Anomalies found during simulation of DCM work.

VI. Conclusions

In this paper, we addressed the problem of cleaning sensor data in smart heating control systems. We presented a case of the Smart Heating Control System (SHCS) installed in the South Ural State University, and described the structure and development principles of Data Cleaning Module (DCM) of SHCS. We developed DCM for each single sensor of SHCS as a set of subsystems, namely Preprocessor, Predictor, Reconstructor, and Anomaly Detector.

Preprocessor prepares a training set to learn Predictor's Recurrent Neural Network (RNN). Preprocessor includes the following subsystems. Parser extracts a part of the sensor data from the data warehouse, and transforms it in an appropriate way. Next, Restorer imputes missing values in the parsed data. Then, Outlier Detector finds deviant data points, substitutes them to NULLs as if they missed, and Restorer imputes once again. Finally, Normalizer splits imputed sensor data into a set of normalized subsequences.

Predictor provides RNN with long short-term memory (LSTM) layer learned on the data prepared by Preprocessor. When using, RNN takes a subsequence of the real sensor values preceding to the predicted value as an input, and outputs the predicted value.

Reconstructor takes the current non-NULL value of the sensor, and checks if it is significantly dissimilar to the value produced by Predictor. If so, the real value is substituted by the synthetic value from Predictor.

Anomaly Detector takes a set of subsequences in the time series of the sensor data ended by the current value, checks if a subsequence is anomalous, and if so, notifies the operator of SHCS.

We presented experimental evaluation of DCM on the real data from sensors of SHCS, and experiments illustrated that proposed approach are able to achieve higher accuracy.

In further studies, we plan to elaborate our approach by application of parallel algorithms for mining time series data [25], [26].

REFERENCES

[1] M. Abadi, P. Barham, J. Chen, and Z. Chen, A. Davis, "TensorFlow: A system for large-scale Machine Learning," OSDI, pp. 265–283, 2016. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

[2] T. Ahmad, H. Chen, and Y. Huanga, "Short-term energy prediction for district-level load management using machine learning based approaches," Energy Procedia, vol. 158, pp. 3331–3338, 2019. DOI: 10.1016/j.egypro.2019.01.967

[3] Anomaly Detection Toolkit, User Guide. [Online]. Available: https://arundo-adtk.readthedocs-hosted.com/en/stable/userguide.html

[4] A. Van Benschoten, A. Ouyang, F. Bischoff, and T. Marrs, "MPA: a novel cross-language API for time series analysis," Journal of Open Source Software, vol. 5, no. 49, 2020. DOI: 10.21105/joss.02179

[5] A. Basalaev, "Automated energy management for heat and power system of university campus," Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics, vol. 15, no. 4, pp. 26–32, 2015. [Online]. Available: https://vestnik.susu.ru/ctcr/article/view/4355

[6] A. Basalaev, M. Tochilkin, and D. Shnayder, "Enhancing room thermal comfort conditions modeling in buildings through schedule-based indicator functions for possible variable thermal perturbation inputs," Proc. of 2019 Int. Conf. on Industrial Engineering, Applications and Manufacturing, pp. 1–8, 2019. DOI: 10.1109/ICIEAM.2019.8742907

[7] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time series analysis: forecasting and control. Whiley, 2015.

[8] R. Dunia and S. Joe Qin, "Joint diagnosis of process and sensor faults using principal component analysis," Control Engineering Practice, vol. 6, is. 4, pp. 457–469, 1998. DOI: 10.1016/S0967-0661(98)00027-6

[9] Sh. Farouq, S. Byttner, M.-R. Bouguelia, and N. Nord, "Large-scale monitoring of operationally diverse district heating substations: A reference-group based approach," Eng. Appl. Artif. Intell., vol. 90, pp. 1–16, 2020. DOI: 10.1016/j.engappai.2020.103492

[10] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735

[11] Y. Hu, H.Chen, G. Li, and H. Li, "A statistical training data cleaning strategy for the PCA-based chiller sensor fault detection, diagnosis and data reconstruction method," Energy and Buildings, vol. 112, pp. 270–278, 2016. DOI: 10.1016/j.enbuild.2015.11.066

[12] S. Idowu, S. Saguna, C. Åhlund, and O. Schelén, "Applied machine learning: Forecasting heat load in district heating system," Energy and Buildings, vol. 133, pp. 478–488, 2016. [Online]. Available: https://DOI.org/10.1016/j.enbuild.2016.09.068

[13] K. Jha, "Minimal loop extraction for leak detection in water pipe network," Proc. of 2012 1st Int. Conf. on Recent Advances in Information Technology, pp. 687–693, 2012. DOI: 10.1109/RAIT.2012.6194578

[14] E. J. Keogh, J. Lin, and A. W. Fu, "HOT SAX: efficiently finding the most unusual time series subsequence," Proc. of the 5th IEEE Int. Conf. on Data Mining, pp. 226–233, 2005. DOI: 10.1109/ICDM.2005.79

[15] E. J. Keogh, S. Lonardi, and C. A. Ratanamahatana, "Towards parameter-free data mining," Proc. of the 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 206–215, 2004. DOI: 10.1145/1014052.1014077

[16] Keras Developer Guides. [Online]. Available: https://keras.io/guides/

[17] P. Malhotra, L. Vig, G. M. Shroff, and P. Agarwal, "Long Short Term Memory Networks for anomaly detection in time series," Proc. of the 23rd European Symposium on Artificial Neural Networks, pp. 89–94, 2015. [Online]. Available: http://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2015-56.pdf

[18] W. J. N. Turner, A. Staino, and B. Basu, "Residential HVAC fault detection using a system identification approach," Energy and Buildings, vol. 151, pp. 1–17, 2017. DOI: 10.1016/j.enbuild.2017.06.008

[19] V. Venkatasubramanian, "Process Fault Detection and Diagnosis: Past, Present and Future," IFAC Proc. Volumes, vol. 34, is. 27, pp. 1–13, 2001. DOI: 10.1016/S1474-6670(17)33563-2

[20] K. Yan, Zh. Ji, and W. Shen, "Online fault detection methods for chillers combining extended Kalman filter and recursive one-class SVM," Neurocomputing, vol. 228, pp. 205–212, 2017. DOI: 10.1016/j.neucom.2016.09.076

[21] D. Yankov, E. J. Keogh, and U. Rebbapragada, "Disk aware discord discovery: finding unusual time series in terabyte sized datasets," Knowl. Inf. Syst., vol. 17, no. 2, pp. 241–262, 2008. DOI: 10.1007/s10115-008-0131-9

[22] Y. Zhao, T. Li, X. Zhang, and C. Zhang, "Artificial intelligence-based fault detection and diagnosis methods for building energy systems: Advantages, challenges and the future," Renewable and Sustainable Energy Reviews, vol. 109, pp. 85–101, 2019. DOI: 10.1016/j.rser.2019.04.021

[23] Y. Zhao, C. Zhang, Y. Zhang, and Z. Wang, "A review of data mining technologies in building energy systems: Load prediction, pattern identification, fault detection and diagnosis," Energy and Built Environment, vol. 1, is. 2, pp. 149–164, 2020. DOI: 10.1016/j.enbenv.2019.11.003

[24] N. Zimmerman, E. Dahlquist, and K. Kyprianidis, "Towards on-line fault detection and diagnostics in district heating systems," Energy Procedia, vol. 105, pp. 1960–1966, 2017. DOI: 10.1016/j.egypro.2017.03.567

[25] M. Zymbler and Ya. Kraeva, "Discovery of time series motifs on intel many-core systems," Lobachevskii Journal of Mathematics, vol. 40, no. 12. pp. 2124–2132, 2019. DOI: 10.1134/S199508021912014X

[26] M. Zymbler, A. Polyakov, and M. Kipnis, "Time series discord discovery on intel many-core systems," Revised Selected Papers. Communications in Computer and Information Science, vol. 1063. pp. 168–182, 2019. DOI: 10.1007/978-3-030-28163-2_12