sciendo

Autonomous Sensor Data Cleaning in Stream Mining Setting

Klemen Kenda, Dunja Mladenić
Jožef Stefan Institute, Ljubljana, Slovenia
Jozef Stefan International Postgraduate School, Ljubljana, Slovenia

Abstract

Background: Internet of Things (IoT), earth observation and big scientific experiments are sources of extensive amounts of sensor big data today. We are faced with large amounts of data with low measurement costs. A standard approach in such cases is a stream mining approach, implying that we look at a particular measurement only once during the real-time processing. This requires the methods to be completely autonomous. In the past, very little attention was given to the most time-consuming part of the data mining process, i.e. data pre-processing. **Objectives:** In this paper we propose an algorithm for data cleaning, which can be applied to real-world streaming big data. **Methods/Approach:** We use the short-term prediction method based on the Kalman filter to detect admissible intervals for future measurements. The model can be adapted to the concept drift and is useful for detecting random additive outliers in a sensor data stream. **Results:** For datasets with low noise, our method has proven to perform better than the method currently commonly used in batch processing scenarios. Our results on higher noise datasets are comparable. **Conclusions:** We have demonstrated a successful application of the proposed method in real-world scenarios including the groundwater level, server load and smart-grid data.

Keywords: big data, autonomous processing, real-world applications, data cleaning, stream mining, water management, data-centre management, smart-grids
JEL classification: C55, C81, C63, C67
Paper type: Research article

Received: Jan 31, 2018
Accepted: Apr 21, 2018

Citation: Kenda, K., Mladenić, D. (2018), "Autonomous Sensor Data Cleaning in Stream Mining Setting", Business Systems Research, Vol. 9, No. 2, pp. 69-79.
DOI: 10.2478/bsrj-2018-0020

Acknowledgments: This work was supported by the Slovenian Research Agency and the ICT program of the EC under project OPTIMUM (H2020-MG-636160) and Water4Cities (H2020-MSCA-RISE-734409).

Introduction

Big Data is a term that is used for datasets that are too large in size and complexity to be handled with the current methodologies (Fan et al., 2013). The meaning of this definition changes constantly with the development of technology and advances in computer science. However, translating the data analysis into a streaming on-line

process is always considered a good approach. Stream mining exposes another benefit of the methodology - real-time responsiveness of the system, which has been identified as desirable by many different authors regarding reporting (Belfo et al., 2015), intrusion detection (Al Quhtani, 2017) and others.

The field has received a lot of attention. Many stream modelling (regression, classification, clustering etc.) and evaluation methods have been developed. However, some data mining process phases as identified in the cross-industry standard process for data mining (CRISP-DM) methodology (Shearer, 2000), have been left aside (Kandel et al., 2011; Krempl et al., 2014). One of those phases, which data cleaning is a part of, is "Data preparation" and is crucial for real-world data mining applications (Zekić-Sušac et al., 2015).

Even in classical data mining task, where all the data is available beforehand, the practitioners claim that data preparation takes up to 80% of the time (Press, 2016). A lot of work is done manually. In stream mining scenario there is no possibility for a constant human intervention, all the data pre-processing needs to be completely autonomous.

Data cleaning represents the first step in data pre-processing. It represents a permanent challenge in data analytics. If not done or badly performed it can result in inaccurate predictions and later in unreliable business decisions. The issue has been tackled recently both by industry and academia, mostly to address the issues of scalability (Big Data), interfaces, new abstractions and statistical techniques (Chu et al., 2016).

The field of time-series analysis has been lively for a number of decades. Kalman published his work on linear filtering already in 1960 (Kalman, 1960). Kalman stands out of the *crowd* due to the successful application of the equations to trajectory estimation in the NASA Apollo space program. Different applications have been reported since then and the field of time-series analysis has been *reinvented* in correspondence with advances in computer science and technology. In the last years many applications were created for on-line streaming data analysis.

Outlier detection in time series has been thoroughly discussed already in 1993 by Chen and Liu (1993). The paper identifies five different types of time series outliers: (1) Additive Outlier (AO), (2) Innovation Outlier (IO), (3) Level Shift (LS), (4) Temporary Change (TC) and (5) Seasonal Level Shift (SLS). Authors propose usage of different models from ARIMA family (AR, MA, IMA, Seasonal IMA) for outlier detection, using its short-term prediction capabilities.

To the best of our knowledge the usage of Kalman filter for cleaning of streaming sensor data has firstly been proposed in our work (Kenda et al., 2013). The paper proposed an algorithm for additive outlier detection in a stream mining setting using short-term prediction based on Kalman filter. The very same idea has been proposed in (Xu, 2015), where it has been studied in depth and extended to a wider context. The authors coined the methodology as time series Kalman filter (TSKF). The method has been improved in (Kenda et al., 2017), where we proposed the usage of unsupervised machine learning approach for automatic parameter fine-tuning and tested the method on an artificial data set. In the current work we further extend the methodology by introducing the indirect modelling-based evaluation procedure and extensive testing on 5 real-world data sets.

Recently, literature is examining other potential Kalman filter extensions for data cleaning. For example, (Marczak et al., 2018) studies usability of augmented Kalman filters (AKF).

The paper is structured as follows. "Methodology" section describes Kalman filter algorithm and how it was implemented in our methodology. In the "Results" section

we provide evaluation of our methodology on artificial and real-world datasets. We also describe the indirect evaluation procedure. Next, we discuss the usability of our methodology in real-world scenarios and compare it to current state-of-the-art in batch setting. Finally, we conclude the paper.

Methodology

The notion of additive outlier

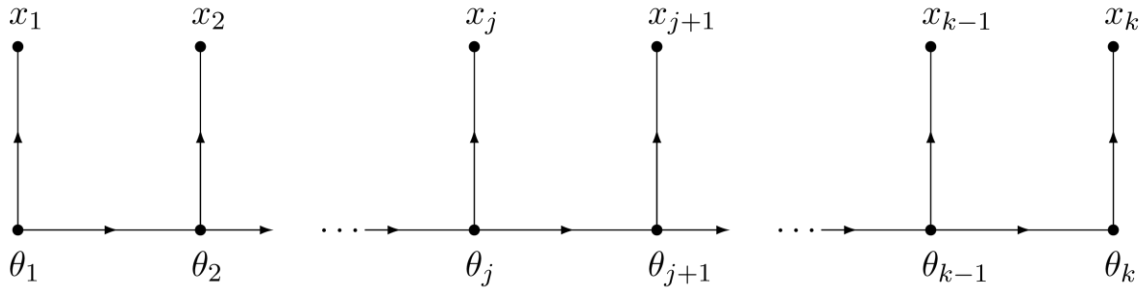
Additive outlier is a point outlier, which occurs at a given timestamp t_j and affects a single observation. In sensor data such outliers can be a consequence of a sudden change in ambient conditions, communication glitch or some similar unexpected event. With sensor measurements we assume that they arrive much faster than the data changes.

We propose a method with short-term prediction, based on previous measurements. Short term prediction is compared to the new measurement and classified as an outlier if the difference exceeds a specified threshold. As proposed in (Kenda et al., 2013) we introduce a safe guard to overcome a potential instability of the algorithm and enlarge the threshold in case that the detected outlier is a false positive, which might be an indication of a sudden concept drift in the data.

Kalman Filter

Kalman filter is a very suitable algorithm to be applied to data cleaning in a streaming scenario. It is an on-line algorithm that can produce short term predictions and even calculate covariance error matrix (used to calculate a threshold for outlier classification). Algorithm assumes that our process can be described as a Gauss-Markov process.

Figure 1
Diagram of Gauss-Markov process

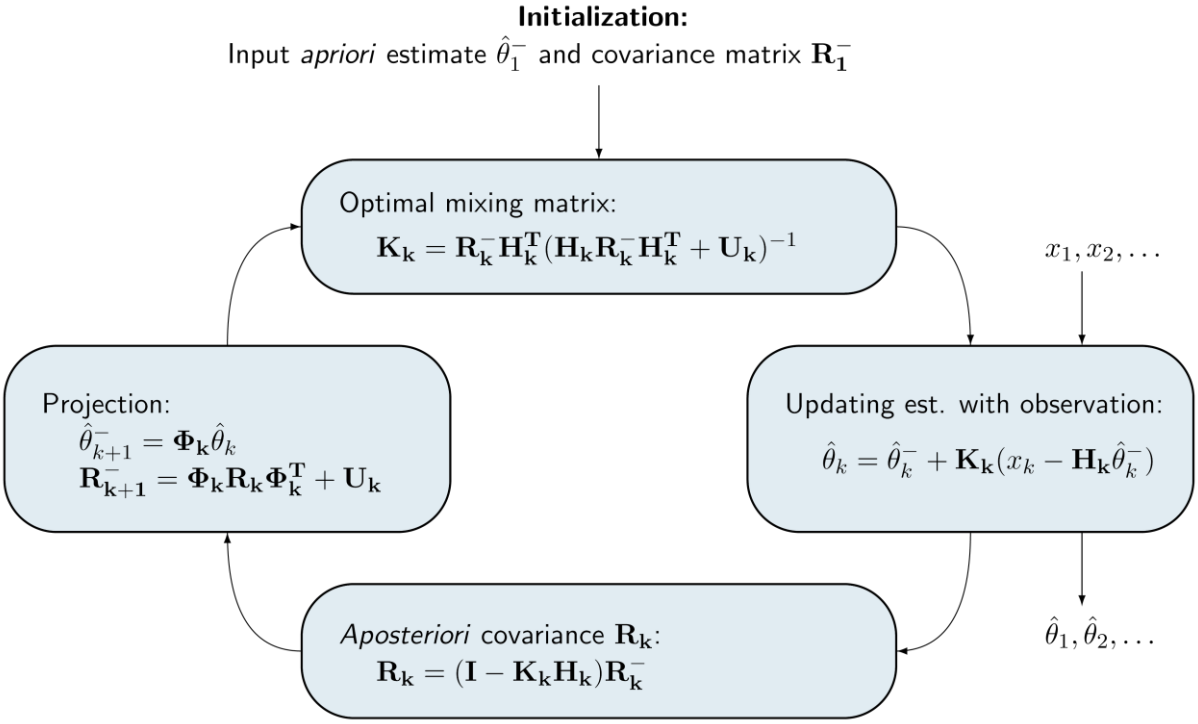


Source: (Kenda et al., 2017)

The process is depicted in Figure 1. Arrows from internal state θ_j to another internal state θ_{j+1} depict transitions (transition equation) and arrows from internal state θ_j to observation x_j depict observation equations. The process has two properties:

- Every consequent internal state θ_{j+1} only depends on a prior internal state θ_j. Both states are connected through transition matrix $\boldsymbol{\Phi_j}$.
- Each internal state θ_j can be inferred through its observation x_j, which is linked to the internal state via observation matrix $\boldsymbol{H_j}$ and is a subject of Gaussian noise.

Figure 2
Kalman filter application cycle



Source: (Kenda, Mladenić, 2017)

In general, matrices $\boldsymbol{H_j}$ and $\boldsymbol{\Phi_j}$ can change over time, but in our case they remain the same as we assume the underlying process does not change through time. Kalman filter equations are depicted in Figure 2.

Kalman filter application cycle starts with initialization of *a priori* estimates for internal state θ_1^- and covariance matrix $\boldsymbol{R_1^-}$. With each new observation x_j the state and covariance matrix get updated. The next phase is dedicated to short-term one step ahead prediction (projection). Finally, optimal new mixing matrix gets calculated (responsible for optimal updating of the projected state with an observation). $\boldsymbol{U_k}$ represents normal distribution variance noise matrix.

Computational complexity of our implementation of Kalman filter is $O(n^3)$ where n is the dimension of internal state space. In the proposed 2nd degree model the number of internal state components is $n = 3$.

Parameter Learning

Initialization of Kalman filtering algorithm can be very demanding and there can be many free parameters involved, depending on the observation and transition matrix dimensions. Usage of expectation maximization (EM) algorithm (Dempster et al., 1977; Xu, 2015) can yield estimates for the initial internal state of the system and corresponding covariance matrices. Clean initial dataset is needed to obtain these parameters.

In our experiments with time series data the results from EM algorithm have not provided good results (confidence into last state was exaggerated), therefore we propose an additional data-oriented approach. EM calculates estimates of the following parameters: *a priori* initial state θ_1^-, transition covariance $\boldsymbol{Q}$, observation covariance $\boldsymbol{R_k}$ and initial state covariance $\boldsymbol{R_1^-}$. We propose multiplying EM estimates with an additional factor in order to minimize F_1 score of outlier classification on a

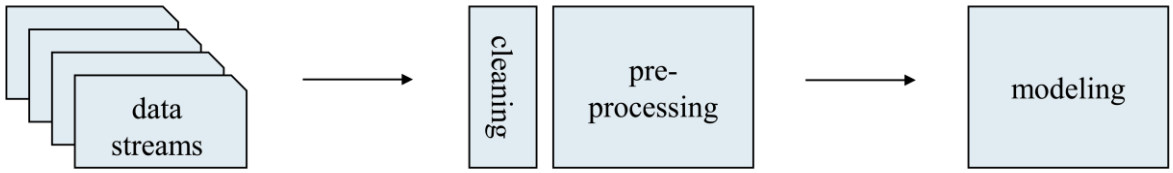
labelled dataset. Parameters can be obtained by a grid search over a predefined multiplier space.

Grid search is time consuming, but it can find configurations which result in much smoother model that better follows the underlying dynamic processes in the data. We have implemented exhaustive and randomized grid searches in our solution, reported results are based on the randomized version.

Streaming Sensor Data Platform with Data Cleaning

We propose the usage of the filter at the lowest possible level in the pre-processing platform. The data-cleaning component should be implemented at the entry point of a particular data source to the pre-processing platform (see Figure 3). Clean data is then inserted into stream pre-processing engine, which is in charge of data enrichment and heterogeneous data fusion and finally this data is pushed into the appropriate stream modelling method. Cleaning at this level uses only autoregressive features. On a higher level, however, data-cleaning, which takes advantage of data fusion, could be used.

Figure 3
Position of data-cleaning system within the stream-mining analytical platform



Source: (Kenda, Mladenić, 2017)

Results

We tested our results on artificial and real-world data sets. Functionality of the algorithm is illustrated in Figure 4. It shows the impact of Kalman filter's short-term prediction and its variance on additive outlier detection. The measurement (depicted in dark blue) that falls outside the admissible interval around short term prediction (depicted in light blue) is considered an outlier.

Results on Annotated Artificial Data Set

We provide an artificial dataset, following the usual daily profile of a family of typical sensors. Each time-series in the dataset introduces a different level of Gaussian noise $N(\mu = 0; \sigma)$. We have made the dataset publicly available at ResearchGate (Kenda, 2017). Data points are a subject of noise, 1% of data points have been considered as candidates for an additive outlier. Amplitude of additive outliers has been uniformly sampled on the interval from 0 to $0.714 \cdot \max(f(t))$, where $\max(f(t))$ is the maximum value of the underlying dynamics function. Amplitudes that were lower than $2 \times \sigma$ have been dismissed.

Artificial set experimental results are depicted in Table 1. Different data sets (from 1 to 9) introduce different Gaussian noise, which makes it more and more difficult to correctly classify the outliers, which can be observed in decreasing values of precision, recall and F_1 in Table 1. As expected, ARIMA (batch) method gives slightly better results than Kalman (streaming) method. F_1 scores are similar, whereas ARIMA method is optimized towards better precision and Kalman towards better recall.

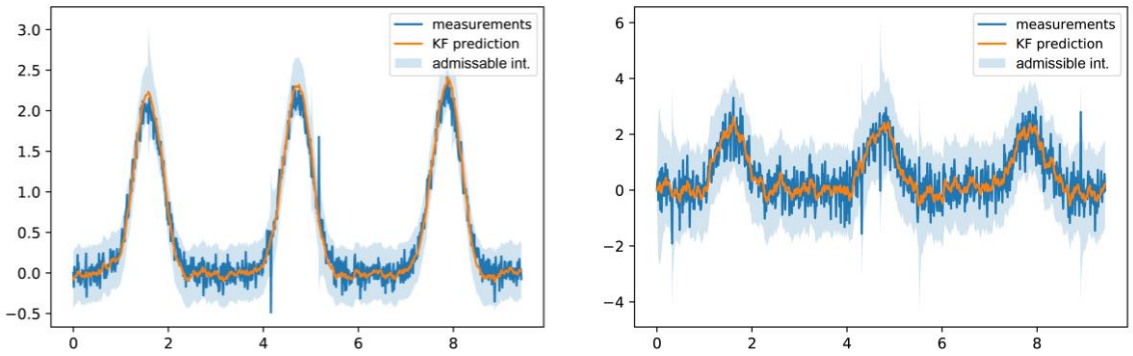
Figure 4 shows algorithm results with 2 different datasets: left - little noise ($\sigma = 0.036$), right - more noise ($\sigma = 0.179$). Kalman filters' short-term prediction is depicted in orange, measurements in dark blue. Any measurement outside of the admissible light-blue interval (defined by Kalman filter variance) is considered as an outlier.

Table 1
Comparison of Kalman filter additive outlier detection results with current batch methodology (Chen et al., 1993)

		Kalman filter method			ARIMA method		
Dataset	**Noise σ**	**Precision**	**Recall**	**F_1**	**Precision**	**Recall**	**F_1**
1	0.036	0.866	0.967	0.914	0.624	0.874	0.728
2	0.071	0.776	0.983	0.867	0.940	0.829	0.881
3	0.107	0.737	0.872	0.799	0.906	0.750	0.821
4	0.143	0.681	0.946	0.792	0.944	0.740	0.830
5	0.179	0.695	0.592	0.640	0.902	0.643	0.751
6	0.213	0.455	0.873	0.598	0.896	0.520	0.658
7	0.250	0.587	0.373	0.456	0.790	0.448	0.571
8	0.286	0.435	0.779	0.558	0.816	0.461	0.589
9	0.321	0.353	0.545	0.428	0.741	0.336	0.462

Source: (Kenda et al., 2017).

Figure 4
Illustration of the algorithm results with 2 different datasets: lower noise (left) and higher noise (right); measurements outside the admissible intervals are detected as outliers



Source: (Kenda, Mladenić, 2017)

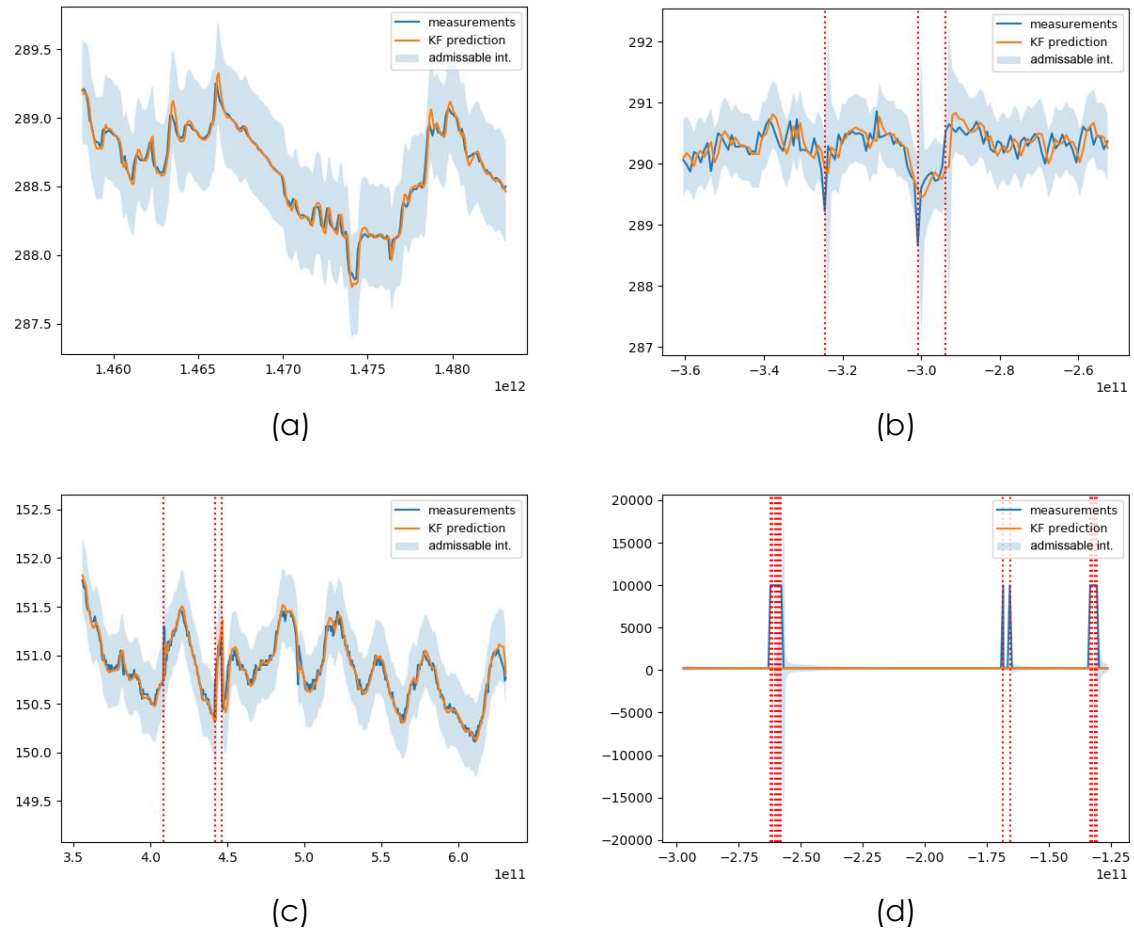
Results on Real-world Data Sets

There are two major problems concerning real-world sensor data sets: (1) these data sets are not annotated, therefore it is impossible to calculate proper accuracy measures of a data cleaning algorithm, (2) without accuracy measures it is also impossible to apply machine-learning techniques for parameter learning.

To overcome these shortcomings, we need to take a look into characteristics of sensor data. We have observed in many sensor data sources that outliers are rare. Most of the data is clean. It is therefore easy to introduce artificial outliers into original data and use such augmented data set to solve the problem (2). With the algorithm we are able to learn adequate parameters for a successful application of the algorithm. Solving problem (1) is more difficult. We can apply human-based anomaly classification for the rare detected outliers, which enables us to calculate

precision (is detected outlier really an outlier?). The second method is to compare modelling performance (i.e. regression) between the clean and the original datasets.

Figure 5
Illustration of the algorithm results with underground water level dataset: (a) time-series without outliers, (b) and (c) time-series with true and false positive outliers, (d) time-series with obvious outliers



Source: Authors' work

We have analysed performance of our method on 340 time-series data sets of groundwater levels from Slovenia. Results are depicted in Figure 5. Y-axis depicts groundwater levels in meters above sea level, x-axis depicts unix timestamp. Figure 5(a) shows a smooth and clean time series, which is easy to model with Kalman filter. The algorithm successfully identifies even bigger shifts in the groundwater levels. Figures 5(b) and (c) show sensors with more noise. The timestamps where potential outliers were detected are marked with a vertical red dotted line. We can observe two true positives (first two outliers) and one probable false positives in Figure 5(b), which is a consequence of a fast change in the data and is difficult to model in an on-line setting. Similarly, we can notice one true and two false positives in Figure 5(c). Figure 5(d) depicts extreme errors in the data that get detected correctly, even in cases, where there is more than one consecutive noisy measurement present.

Indirect Evaluation of Data Cleaning with Modelling Results

Without a labelled dataset from real-world scenarios, we cannot directly estimate the effect of data cleaning. Thus we are estimating the benefits of data cleaning through observation of the improvements of machine learning models on the data. It has been previously shown that data cleaning can significantly improve the model accuracy (Krishnan et al., 2016). We have compared root mean squared error (RMSE) of ARIMA (1, 1, 0) models on raw and on cleaned datasets. Lower RMSE measure means better fit of the models to the dataset.

Furthermore, we have developed a meta-classification algorithm for time-series to detect suitable candidates, where RMSE can be improved. Based on the meta-data obtained from the time-series (such as variance, mean data frequency, Kalman filter parameters, confidence of the Kalman model, etc.) and from the data cleaning algorithm learning phase, such as (learning parameters, number of errors, length of data frame and cleaning model score), we were able to build a classifier, which can predict whether our cleaned time-series can be modelled worse, better or equally good on cleaned data. The classifier has been built using the random forests algorithm (Breiman, 2001).

Experiments have been conducted on 5 different datasets: (i) 340 time-series of groundwater levels in Ljubljana region, (ii) 67 time-series from Yahoo! A1 Server Load (Yahoo! Webscope, 2015), (iii) 400 time-series from smart-grid observations (active power) in SW Slovenia, (iv) and (v) 100 synthetic time-series from Yahoo! anomaly detection benchmark. Results are depicted in Table 2. Table presents KPIs related to the algorithm and the meta-classifier performance as follows. *Improvement* indicates fraction of time-series with better fit after cleaning (0.805 means that 80.5% of time-series benefited from the proposed data cleaning). RMSE ratio expresses ratio of improvements of RMSE against the losses (443.6 indicates that RMSE is improved much more than it deteriorates in cases, where data cleaning fails; this happens as groundwater data contains significant human-made errors). Precision, recall and F_1 are standard classifier evaluation measures for our meta-classification algorithm.

Table 2

Algorithm performance on unlabelled data and prediction of the meta-classifier regarding the success of the algorithm

	Algorithm performance		Classification performance		
Dataset	**Improvement**	**RMSE ratio**	**Precision**	**Recall**	**F_1**
Groundwater	0.513	443.6	0.737	0.737	0.737
Server load	0.530	1.400	0.746	0.740	0.739
Smart-grid	0.805	1.270	0.850	0.861	0.850
Yahoo! A2 (synthetic)	1.000	N/A	1.000	1.000	1.000
Yahoo! A3 (synthetic)	0.000	N/A	N/A	N/A	N/A

Source: Authors' work

The most illustrative are results on the two synthetic datasets. On the first dataset (Yahoo! A2) our algorithm works perfectly, while on the second dataset (Yahoo! A3) it fails completely. The main difference between these two datasets is that the periodicity in the first dataset is much larger and noise is much lower. The same properties are illustrated on real-world datasets, where we see the best performance (80.5%) of the algorithm on a smart-grid dataset. Typical period in this dataset is one day and measurements are taken every 15 minutes. Groundwater (i) and server load (ii) datasets have a sampling interval much closer to the typical period (significant

change in the data can happen within a single sampling interval, i.e. groundwater can rise significantly in a day with substantial amount of rainfall). Performance of our algorithm is 51.3% and 53.0%, respectively.

Usability of the cleaning algorithm was further improved with a meta-classifier. Based on time-series metadata the classifier is able to identify the data sources which are likely to improve with our algorithm with a precision, that is much higher than the improvement ratio (between 73.7% and 85.0%).

Discussion

As presented in the previous section our algorithm achieves the best performance with a typical stream of sensor data, as we can find in Internet of Things. In such scenarios sensor measurements are frequent and systematic changes in the data are low (sampling interval is much shorter than periodicity). In comparison with a commonly used ARIMA methodology in batch data pre-processing (Chen et al., 1993), our method works better with lower noise data. An obvious downside of the ARIMA methodology is that it requires fitting of ARIMA model to the whole dataset, which makes it unusable with data streams.

Our approach is applicable in any kind of streaming scenario. However, there are some additional restrictions that need to be considered. When testing on real-world dataset we have observed heterogeneous characteristics of sensor data with respect to noise, volatility and measurement intervals. When dealing with large and diverse amounts of sensors (nowadays it is not unusual to have more than 10.000 sensors in the system, i.e. in a regional smart-grid system) it is not feasible to do individual cleaning model learning, therefore some basic clustering of sensors into groups with similar properties is needed. Fine tuning of the parameters can be performed on a representative time-series only and then applied to the whole cluster.

Based on their characteristics efficiency of our methodology differs between the datasets. However, efficiency of the algorithm can be further improved with a classification algorithm on the top of time-series/learning-phase metadata, which is able to select a suitable time-series for the data-cleaning algorithm. In this way we were able to achieve precisions between 73-85%.

Conclusion

In this paper we have identified that efficient data pre-processing is very important in streaming data scenarios. We have focused on the first part of the data pre-processing pipeline: data cleaning. We conducted a short research on the state-of-the-art in the field and proposed our own method based on Kalman filter. The method has been quantitatively tested on an artificial data set. We have compared our method to the ARIMA state-of-the-art method and have obtained better results on the datasets with lower noise ratio and comparable results on the datasets with higher noise ratio. The main advantage of our method is, that it can work with Big Data in a streaming scenario.

Additionally, we have applied our method to a heterogeneous set of real-world time-series. We have tested the efficiency of our cleaning method with an indirect approach, where we tried to fit an ARIMA model to raw data and to clean data to compare the respected error measures. The proposed data cleaning was shown to be beneficial on time-series that have properties like majority of sensor streams available in the IoT domain. We also developed a meta-classification method which can predict the success of the data cleaning with 75%-85% precision.

By observing differences in Yahoo! A2 and Yahoo! A3 datasets we identified the major limitation of our algorithm. When changes in a time-series are rapid (i.e. if periodicity is short in comparison to measurement frequency) many valid measurements are classified as outliers and algorithm accuracy is low. Future work should therefore be directed into improving Kalman filter parameter fine-tuning procedure, which should capture such behaviour. Additionally, usability of the algorithm should be tested on different real-world datasets and in the production environment.

References

1. Al Quhtani, M. (2017), "Data Mining Usage in Corporate Information Security: Intrusion Detection Applications", Business Systems Research, Vol. 8, No. 1, pp. 51-59.
2. Belfo, F., Trigo, A., Estébanez, R. P. (2015), "Impact of ICT Innovative Momentum on Real-Time Accounting", Business Systems Research, Vol. 6, No. 2, pp. 1-17.
3. Breiman, L. (2001), "Random Forests", Machine Learning, Vol. 45, No. 1, pp. 5-32.
4. Chen, C., Liu, L. (1993), "Joint Estimation of Model Parameters and Outlier Effects in Time Series", Journal of the American Statistical Association, Vol. 88, No. 421, pp. 284-297.
5. Chu, X., Ilyas, I. F., Krishnana, S., Wang, J. (2016), "Data cleaning: Overview and emerging challenges", in Özcan, F., Koutrika, G. (Eds.), Proceedings of the 2016 International Conference on Management of Data, ACM, San Francisco, pp. 2201-2206.
6. Dempster, A. P., Laird, N. M., Rubin, N. M. (1977), "Maximum likelihood from incomplete data via the EM algorithm", Journal of Royal Statistical Society Series B, Vol. 39, No. 1, pp. 1-38.
7. Fan, W., Bifet, A. (2013), "Mining big data: Current status, and forecast to the future", ACM siGKDD Explorations Newsletter, Vol. 14, No. 2, pp. 1-5.
8. Kalman, R. E. (1960), "A new Approach to linear filtering and prediction problem", Journal of basic engineering, Vol. 82, No. 1, pp. 34-45.
9. Kandel, S., Heer, J., Plaisant, C., Kennedy, J., van Ham, F., Riche, N.H., Weaver, C., Lee, B., Brodbeck, D., Buono, P. (2011), "Research directions in data wrangling: Visualizations and transformations for usable and credible data", Information Visualization Journal, Vol. 10, No. 4, pp. 271-288.
10. Kenda, K. (2017), Artificial data-set for testing time-series additive outlier detection methods, available at: https://www.researchgate.net/publication/317721142_Artificial_data-set_for_testing_time-series_additive_outlier_detecion_methods (18 February 2018).
11. Kenda, K., Mladenić, D. (2017), "Autonomous on-line outlier detection framework for streaming sensor data", in Zadnik Strin, L., Kljajić Borštnar, M., Žerovnik, J., Drobne, S. (Eds.), Proceedings of the 14th International Symposium on Operational Research, Bled, pp. 103-108.
12. Kenda, K., Škrbec, J., Škrjanc, M. (2013). "Usage of Kalman Filter for Data Cleaning of Sensor Data", in Gams, M. (Ed.), Proceedings of the 16th International Multiconference Information Society – IS 2013, Ljubljana, pp. 172-175.
13. Krempl, G., Žliobaite, I., Brzezinski, D., Hüllenmeier, E., Last., M., Lemaire, V., Noack, T., Shaker, A., Sievi, S., Spiliopolou, M. (2014), "Open challenges for data stream mining research", ACM siGKDD Explorations Newsletter, Vol. 16, No. 1, pp. 1-10.
14. Krishnan, S., Wang. J., Wu, E., Franklin, M. J., Goldberg, K. (2016), "ActiveClean: interactive data cleaning for statistical modeling", in Chaudhuri, S., Haritsa, J. (Eds.), Proceedings of the VLDB Endowment, Vol. 9, No. 12, pp. 948-959.
15. Marczak, M., Proietti, T., Grassi, S. (2018), "A data-cleaning augmented Kalman filter for robust estimation of state space models", Econometrics and Statistics, Vol. 5, pp. 107-123.
16. Press, G. (2016), "Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says", available at: https://www.forbes.com/sites/suntrustprivatewealth/2017/12/21/wealth-transfer-are-you-sure-your-beneficiaries-are-prepared/ (31 January 2018).
17. Shearer, C. (2000), "The CRISP-DM model: the new blueprint for data mining", Journal of

data warehousing, Vol. 5, No. 4, pp. 13-22.
18. Xu, S. (2015), "Data Cleaning and Knowledge Discovery in Process Data", PhD thesis, University of Texas, Austin.
19. Yahoo! Webscope (2015), "S5 - A Labeled Anomaly Detection Dataset, version 1.0", available at: http://research.yahoo.com/Academic_Relations (28 February 2018).
20. Zekić-Sušac, M., Has, A. (2015), "Data Mining as Support to Knowledge Management in Marketing", Business Systems Research Journal, Vol. 6, No. 2, pp. 18-30.

About the authors

Klemen Kenda is a Ph. D. candidate at Jožef Stefan International Postgraduate School and a researcher at Artificial Intelligence Laboratory at Jožef Stefan Institute in Ljubljana, Slovenia. He received his diploma degree from the Faculty of Mathematics and Physics at University of Ljubljana, with a thesis "Usage of machine learning techniques with analysis of the data from ATLAS detector". His main research interests are in information and communication technologies. He is focused on data pre-processing, data fusion, machine learning and data-driven modelling in the context of streaming big data. He is actively engaged in various science, industrial and start-up projects (H2020, cooperation between research and industry, energy efficiency start-up). The author can be contacted at **klemen.kenda@ijs.si.**

Dunja Mladenić works as a researcher and a project leader at Jožef Stefan Institute, Slovenia, leading Artificial Intelligence Laboratory and teaching at Jožef Stefan International Postgraduate School, University of Ljubljana and University of Primorska. She has extensive research experience in study and development of machine learning, data/text mining, semantic technologies, sensor data analysis methods and their application on real-world problems. She has published papers in refereed journals and conferences, coedited several books, served on program committees of international conferences and organized international events. She serves as a project evaluator of project proposals for European Commission and USA National Science Foundation. She served on the Institute's Scientific Council (2013-2017) as a vice president (2015-2017). She serves on executive board of Slovenian Artificial Intelligence Society SLAIS (as a president (2010-2014)) and as an advisory board member of ACM Slovenija. The author can be contacted at **dunja.mladenic@ijs.si.**