Margarida Costa Instituto Superior Técnico, Universidade de Lisboa
INESC-ID

On the role of recurrent neural networks for anomaly detection in water distribution systems

Margarida Costa Susana Vinga Rui Henriques

***Abstract*—Water is an indispensable resource for society. A major task associated with the proper management of water distribution systems is to detect anomalies to support decision-making and make contingency plans. Interesting anomalies can be bursts, unusual demands, and illegal consumption. Water leakage detection and location is a very difficult problem, due to the lack of information about the water system, and a leak might not be easily detected or confused with other events. The methodology proposed detects anomalies in the water system distributions, with a focus on bursts, through the use of deep learning architectures, in particular, encoder-decoder architectures based on LSTM such as CNN-LSTM, CNN-BiLSTM, and SCB-LSTM. Predictions are adjusted by using a temperature correction model. The water system studied is Quinta do Lago, located in Portugal, Algarve region. Infraquinta is the entity that manages the infrastructures. Experimental results in real data show the pitfalls of the unsupervised anomaly detection task in water distribution systems. It also highlights that the proposed methodology, although yielding some properties of interest, needs to be complemented with additional principles to the targeted end. Finally, it pinpoints meaningful differences between recurrent architectures.**

Time series data analysis Anomaly detection Water management system Recurrent neural network

I. Introduction

Water, once an abundant natural resource, is becoming more valuable due to climate change and over-exploitation. Challenges that will be faced with climate change are droughts, flash floods, higher air temperature, and increased household water demand in the hot season. In this context, it is necessary to make distribution systems more resilient to climate changes and prepare plans to optimize water supply. This can be achieved by proper water supply management.

A Water Distribution System (WDS) collects a large amount of data through sensors (e.g. pressure and flow) which needs to be treated to generate useful information, not only for daily control, operation, and management of systems but also for supporting the current and future planning of the urban water infrastructures. Water Management Entities (WME), also referred to as water utilities, are responsible for the operation of a WDS. Among other services, WMEs provide drinking water and wastewater services (including sewage treatment) to residential, commercial, and industrial sectors of the economy. Typically, public entities operate WDSs.

In WDS, outliers in data might be caused by an unusual household or non-household water consumption, leakages, changes in network system operation, sudden system faults (e.g. breaks in pipelines or service connections), meter malfunction, or problems with the telemetry or supervisory control and data acquisition (SCADA) system.

To reduce the efforts and costs of man labor associated with the identification and characterization of outlier events, there is a vast work in this field on traditional machine learning methods, either in an online or offline manner. Usually, traditional machine learning techniques require separate data pre-processing before training, which tends to be very time-consuming and often requires domain knowledge [5].

Recent deep learning approaches have shown to perform well on raw time series data, eliminating the need for pre-processing. Common deep learning architectures have convolutional layers and recurrent layers. Recurrent neural networks (RNN) have the advantage to persist information for later use in the network. This makes them particularly suited for the analysis of temporal data. Convolutional neural networks (CNN) are appropriate for automating spatial feature extraction from time series raw data through sensory signals. Only recently, it was proposed models based on recurrent neural networks (RNN) for time series anomaly detection [6, 17, 21]. Long-Short Term Memory(LSTM) has gained popularity, due to automatic feature extraction abilities, to represent the relationship between a current event and previous events, and handle of time series problems. Commonly, LSTM is not used alone, they are often used in hybrid models, particularly in encoder-decoder architectures. There is a scarcity of works aiming at detecting anomalies in the water domain. The data to be analyzed is collected by public company Infraquinta that manages the water supply system of Quinta do Lago, a luxury tourist destination in Portugal. The data to be analyzed has a high seasonality in water consumption.

This works aims to compare multiple deep learning architectures, joining the efforts of multiple works, without the need for major preprocessing. These architectures will be used as reconstructions models. Thus, first the models are trained with normal sequence data. If the deviation is bigger than the threshold, then an anomaly is detected. Typically, these models are tested in simple anomaly detection scenarios, where data is labeled, the environment is more static, and there are few different types of anomalies. This research addresses this gap by examining the performance of these models in a complex and dynamic scenario such as anomaly detection in water system networks, in an unsupervised way. It is possible to know when a burst was perceived, or a maintenance activity took place. However, is not possible to know exactly when or where a burst occurred or if another type of event occurred e.g. unusual demand. Furthermore, it is hard to differentiate a consumption from a leak event, or valves opening or closing, due to maintenance activities, from a burst. The target problem is intentionally difficult to attack from this unsupervised stance. Thus, good results are not expected. The main objective of this research is to test whether anomalies characterized by leaks can be isolated or not. This work will be organized as follows. In Section 2, definitions of key topics of water system management and deep learning are introduced. Section 3 offers a compilation of related work. Section 4 describes the proposed solution, including the architectures proposed. Section 5 presents and discusses the results in real data. Section 6 summarizes the main conclusions of this work, as well as future directions.

II. Background

In WDS, leakage through pipes has two major types, *burst* and *background* type leakages [1]. *Burst type leakage* is characterized by quick pressure drop and can be easily detected by the pressure sensors. They often are visually noticed, being reported by public or utility workers, thus the repair time is faster. *Background leakage* concerns the outflow from small cracks or deteriorated joints. They are not characterized by quick pressure drop and are not detectable by measuring instruments. Consequently, they go unreported for a long time.

Leakage in distribution systems can be caused by several different factors [22]. Some examples include bad pipe connections, internal or external pipe corrosion, or mechanical damage caused by excessive pipe load (e.g. by traffic on the road above or by a third party working in the system). Other common factors that influence leakages are ground movement, high system pressure, damage due to excavation, pipe age, winter temperature, defects in pipes, ground conditions, and poor quality of workmanship.

The Supervisory Control and Data Acquisition (SCADA) system is used to manage in real-time the water supply system in a utility. This can be done by monitoring the whole system from water sources to the customer. All the operations are done at the command center allow remote control of the WDS. Examples are to make setpoint changes on distant process controllers, to open or close valves or switches, to monitor alarms due to possible bursts, and to gather measurement information [4].

A. Convolutional Neural Network

A convolutional neural network (CNN) is a class of deep neural networks specialized in processing data with a known grid-like topology [7]. Examples include time-series data, which can be thought of as a 1-D grid taking samples at regular time intervals, and image data, which can be thought of as a 2-D grid of pixels. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data.

B. Recurrent Neural Network

Recurrent Neural Networks (RNN) is a class of neural networks for processing sequential data [7]. RNN differs from standard neural networks by allowing the output of hidden layer neurons to feedback and serve as inputs to the neurons. In this way, the network can use the past to understand the sequential nature of the data.

The notation for the hidden state update is:

$$\mathbf{h}_t = \phi \left(W \mathbf{x}_t + U \mathbf{h}_{t-1} \right) \tag{1}$$

The hidden state at time step t is h_t. It is a function of the input $\vec{x}_t$ at the same time step t, modified by a weight matrix W (like the one we used for feed-forward nets) added to the hidden state of the previous time step, $\vec{h}_{t-1}$, multiplied by its own hidden-state-to-hidden-state matrix U, otherwise known as a transition matrix and similar to a Markov chain. The weight matrices are filters that determine how much importance to accord to both the present input and the past hidden state. The error they generate will return via back-propagation and be used to adjust their weights until error can't go any lower.

The sum of the weight input and hidden state is squashed by the function ϕ which condensates very large or very small values, as well as making gradients workable for back-propagation. Because this feedback loop occurs at every time step in the series, each hidden state contains traces not only of the previous hidden state but also of all those that preceded h_{t-1} for as long as memory can persist.

An increasingly popular type of RNN is Long-Short Term Memory network (LSTM) due to their capacity for classifying, processing, and making predictions based on time series data. LSTM has shown its advantages in numerous applications through dealing with the exploding and vanishing gradient problem and keeping short-term "memory" for a long time [23].

C. Encoder-decoder architectures

An encoder-decoder architecture is a model comprised of two sub-models: one called the encoder that reads the input sequences and compresses it to a fixed-length internal representation, and an output model called the decoder that interprets the internal representation and uses it to predict the output sequence.

III. Related Work

A. Outlier Detection using Deep Architectures

Most deep models based on reconstruction errors are trained on normal sequential data. To detect anomalies, the degree of deviation between observed data and the prediction is then computed. When given an anomalous sequence, it may not be able to reconstruct it well, and hence would lead to higher reconstructions errors compared to normal sequences. One approach has been to calculate the anomaly score either by assuming a normal distribution or non-parametric methods. A threshold is then set, either by extensive search or by considering a normal distribution again. An advantage of this method is that anomalous data are not needed to train the model, which is harder to get. A disadvantage is it depends on the chosen threshold.

Deep or stacked architectures are often used with recurrent units (RNN) or convolutional units (CNN). Graves et al. [16] proposed a stacked LSTM for anomaly detection in time series. A network is trained on normal data and used to forecast.

B. Outlier detection using Encoder-Decoder architectures

Autoencoders also use recurrent units (RNN) [17] or convolutional units (CNN) [11] to perform outlier detection in sequential data such as time series. When the series is inherently unpredictable, the LSTM encoder-decoder can produce superior results in comparison to the LSTM predictor [17]. Sparse autoencoders are also used, either individually or in ensembles [12]. Generally, the most commonly used model is a CNN unit as the encoder and then an LSTM as the decoder (CNN-LSTM) [13].

Lee et. al [14] proposed a Convolutional Neural Networks (CNNs) layer, Bidirectional and Unidirectional Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNNs), which is one of a novel deep architecture named stacked convolutional bidirectional LSTM network (SCB-LSTM). The baseline models for which SCB-LSTM was compared was stacked LSTM and stacked bi-LSTM. SCB-LSTM outperformed these baselines.

Canizo et al. [5] proposes the use of independent CNNs (convolutional heads), to deal with anomaly detection in multi-sensor systems. This architecture is called Multi-head CNN–RNN. Each sensor is addressed individually avoiding the need for data pre-processing and allowing for a more tailored architecture for each type of sensor.

C. Water Distribution Network Data Analysis

One of the disadvantages of the methods applied either in traditional machine learning or deep learning is the need to make a database of interesting patterns or leak labeling [19]. To create the database there is the need for domain knowledge. The database also needs to be large. Another disadvantage is time and resources spent at preprocessing the time series [26, 20]

To avoid the cost and time of labeling data by experts, many approaches use simulated data [18, 10]. Since real data is not always labeled with the target events of interest (such as bursts) and labeling is time-consuming and expensive, many approaches use artificial sequences generated by a software and considered to augment data or train networks (cite). The most used software is EPANET. In this way, there is a certainty if the sequence generated is normal or not. It is preferred to use synthetic data because on real data there is no certainty if the sequence is normal or not [10].

1) Deep architectures: For anomaly detection in WDS, methods based on deep learning have currently been the focus of researchers [8, 26].

Hu et al. in 2019 proposed a hybrid model based on CNN and Bi-LSTM for urban water demand prediction [8]. After the model predicts water consumption, a temperature and holiday correction model is used. This temperature and holiday correction model is based on statistical analysis. It receives 5 days of water usage data and the daily maximum temperature. After the temperature correction model, accuracy improved 1%-3%. After the holiday correction model accuracy improved 2%-5%.

The model CNN-Bi-LSTM was compared with other 5 models: LSTM, bidirectional long-term memory networks (Bi-LSTM), CNN, sparse autoencoder (SAEs), and CNN-LSTM. It was concluded that CNN-Bi-LSTM had less prediction error that the others. The training time of CNN-Bi-LSTM is less than LSTM, Bi-LSTM, CNN, and CNN-LSTM, but larger than SAEs. The training convergence of CNN-Bi-LSTM was set in 125 times, which is smaller than the training times of the other five models.

IV. Method

A. Recurrent Autoencoders as a reconstruction model

Studies have shown that stacked CNN and LSTM architectures can build higher levels of representation of sequence data. An encoder-decoder architecture is a model comprised of two parts: the encoder that reads the input sequences and compresses it to a fixed-length internal representation, and the decoder that interprets the internal representation and uses it to predict the output sequence. The encoder is typically a recurrent and/or convolution layer, followed by a decoder, typically recurrent layers. CNN can capture spatially feature dimension and extracts spatial feature vectors from the input signal as a feature detector. Therefore, CNNs are suitable for being the first layer. Bi-LSTM can fully reflect the long-term historical process and future trend. Bi-LSTMs can use of both forward and backward dependencies. When feeding the input sequence to the Bi-LSTMs, both the spatial correlation in different locations and the temporal dependencies of the feature information can be captured during the feature learning process. In this regard, the Bi-LSTMs are very suitable for being the first or second layer placed after CNN of the proposed model to learn useful information from time series data. When predicting future values, the last layer of the architecture only needs to utilize learned features, namely the outputs from lower layers, to calculate iteratively along the forward direction and generate the predicted values. Hence, an LSTM layer, for capturing forward dependency is a better solution to be the last layer of the proposed model in this paper. A typical neural network has a change in the distribution of network activations due to change in network parameters during training - internal covariate shift [9]. To minimize that a layer of Batch Normalization (BN) is applied. The batch normalization (BN) algorithm tries to normalize the inputs to each hidden layer so that their distribution is reasonably constant during training. This has advantages speeding up convergence, having much higher learning rates, and be less careful about parameter initialization. For each convolution, a Batch Normalization is applied followed by an Activation Layer (Relu). Batch normalization was used. This method worked better than pooling. Models are trained with normal sequences and learn to reconstruct them. When given an anomalous sequence, it may not be able to reconstruct it well, and hence would lead to higher reconstruction errors compared to the reconstruction errors for the normal sequences.

B. Computing likelihood of anomaly

The anomaly score is given by the reconstruction error for each point t_i:

$$\mathbf{a}^{(i)} = \left|\mathbf{x}^{(i)} - \mathbf{x}'^{(i)}\right| \tag{2}$$

where x is the observed time series and $x\prime(i)$ is the reconstructed time series.

If above a certain threshold, the point in a sequence is considered an anomaly.

To calculate that threshold in the validation set, it maximizes F-beta score:

$$F_\beta = \left(1+\beta^2\right) \times P \times R / \left(\beta^2 P + R\right) \tag{3}$$

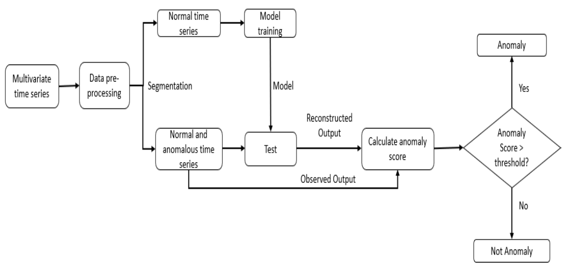

Fig. 1. Pipeline for anomaly detection

C. Data preprocessing

The data is preprocessed, such that does not have missing data or censored data (including left-censored data such as values below detection limits). Raw time series are unevenly spaced. For the research, interpolation was used, so measurements have exact intervals of 1 minute. For the missing values, interpolation was also used. Before training, data is compressed by the MinMaxScaler normalization method, being compressed between the range [-1, 1]. Data was split as train data, normal validation data, v_N, and abnormal validation data, v_A. v_N was used for early stopping. v_A was used to find thresholds. v_A is a sequence of data that has subsequences of normal and abnormal data. v_N has only data considered normal. Normal data is considered to not have reported burst events at the WME database, in the case of real data. The window size is 90 minutes. The number of sequences is 7. In some experiments in real data, seasonality was removed. To remove seasonality from the data, the seasonal component is subtracted from the original series and then differentiates it to make it stationary. To find the ideal percentage of splits, between validation and training, the walk-forward approach was used, using an expanding window. This method is also referred to as a rolling window analysis.

D. Hyperparameter Tuning

Bayesian Optimization using Gaussian Processes was used to find the best set of parameters. The python library used was scikit-optimize. The objective was to minimize validation loss and time expend to train. Blocked Rolling out cross-validation was applied [3]. The optimal parameters found for the encoder-decoder with an initial CNN part are indicated in Table I.

The optimal parameters found for the stacked architectures are indicated in Table II.

Metrics	Size
Stride size	2
Batch size	64
Encoder layers	2
Decoder layers	3
Number Kernels	5
Number filters	16
Dropout rate	0.25

TABLE I
PARAMETERS FOR CNN

Metrics	Size
Batch size	128
Stacked layers	3
Learning rate	0.01

TABLE II
PARAMETERS FOR STACKED ARCHITECTURES

V. EXPERIMENTS

Real data is a much more dynamic scenario, where is not possible to know exactly where or when a leak occurred or even if an event of another type occurs. Only burst events reported in the database by workers in the WME database were considered. These reported events are only burst related or some other maintenance operation that required open and close valves. Only flow sensors were considered.

Table V compares training time and error of the last trained epoch of 5 different models.

Results	Validation error	Training error	Training time (s)	Epochs
CNN-LSTM	0.0471	0.0427	224	17
CNN-BiLSTM	0.0331	0.0400	215	11
SCB-LSTM	0.0656	0.0652	136	19
Stacked BiLSTM	0.00527	0.005207	3314	21
Stacked LSTM	0.0215	0.0146	3548	26

TABLE III
ANALYSIS OF CONVERGENCE FOR ALL ARCHITECTURES FOR REAL DATA

Results	Accuracy	Precision	Recall	True positive rate	False positive rate
CNN-LSTM	0.91	0.39	0.014	0.74	0.98
CNN-BiLSTM	0.91	0.32	0.023	0.54	0.98
SCB-LSTM	0.91	0.42	0.006	0.85	0.99
Stacked BiLstm	0.91	0.51	0.13	1.0	0.87
Stacked LSTM	0.91	0.36	0.09	0.64	0.92

TABLE IV
RESULTS FOR REAL DATA FOR 1 SEPTEMBER TO 30 DECEMBER OF 2017 FOR 6 BURSTS

Stacked BiLSTM had the lowest training and validation error. For that, it trades training time. Stacked architectures are the ones with the longest training time. Convolutional layers as encoders are very fast to converge, but its SCB is the fastest. The best model to detect leaks considered was CNN-BiLSTM, even though stacked Bi-LSTM was the one with the lowest loss.

Removing seasonality made the train time to be shorter. Loss was overall smaller too.

When seasonality was removed, results got worse for detecting leaks.

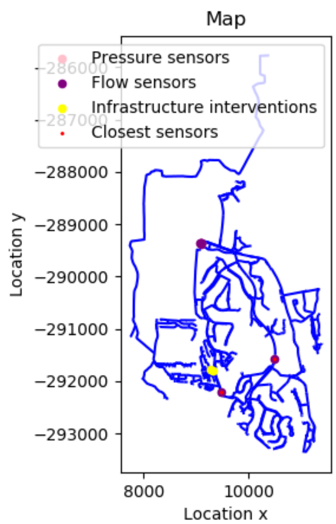

Fig. 2. Map of WDS with marked sensors and possible location of leak.

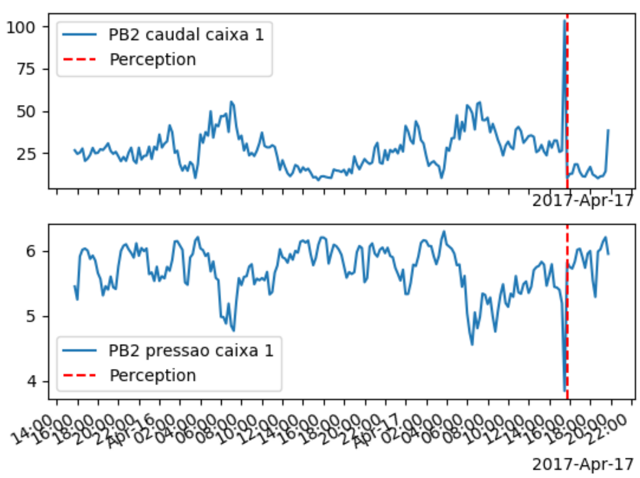

Fig. 3. Time series of flow and pressure sensors considered as close in Figure 2. In red, time of perception reported by the WME.

Results	Validation error	Training error	Training time (s)	Epochs
CNN-LSTM	0.0395	0.0389	152	8
CNN-BiLSTM	0.0331	0.0337	138	17
SCB-LSTM	0.0352	0.0445	123	17
Stacked BiLstm	0.00509	0.00473	2408	21
Stacked LSTM	0.0127	0.0111	2886	21

TABLE V
RESULTS FOR REAL DATA REMOVING SEASONALITY

VI. CONCLUSION

In this research, architectures could effectively extract the characteristics of water flow. It also was analyzed their ability to detect anomalies in a dynamic and complex scenario. The best model was considered CNN-BiLSTM, even though Stacked BiLSTM was the one with the lowest loss. This research was able to analyze differences between techniques and

Results	Accuracy	Precision	Recall	True positive rate	False positive rate
CNN-LSTM	0.83668	0.0806	0.0692	0.836	1.014
CNN-BiLSTM	0.7439	0.0886	0.182	0.925	1.018
SCB-LSTM	0.749	0.0839	0.164	0.872	1.030
Stacked BiLSTM	0.820	0.0940	0.106	0.988	1.0013
Stacked LSTM	0.390	0.0960	0.64	1.01	0.97

TABLE VI
RESULTS FOR REAL DATA REMOVING SEASONALITY FOR 1 SEPTEMBER TO 30 DECEMBER OF 2017 FOR 6 BURSTS

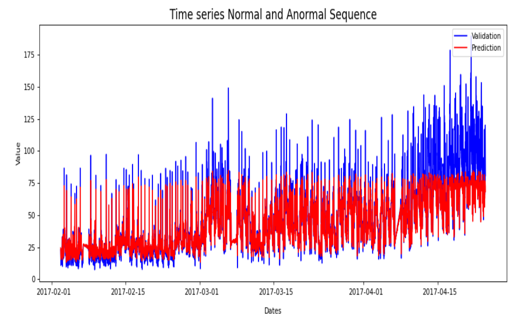

Fig. 4. Comparison between the original and reconstructed sensor readings predicted on an abnormal validation set by a CNN-BiLSTM

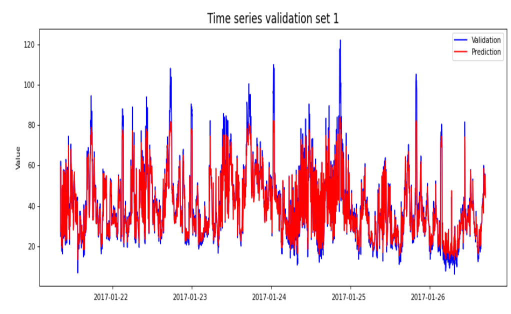

Fig. 5. Comparison between the original and reconstructed sensor readings predicted on a normal validation set by a CNN-BiLSTM

it also identified the problems and this specific scenario. The idea of AEs is straightforward and generic, even for complex data as this scenario. Different types of AE variants can be built to perform anomaly detection. It has reasonable results for a small dataset comparing with the sizes of the datasets of other works. It can also perform promising performance with only raw data by itself (e.g. not major preprocessing). However, autoencoders also have disadvantages. There is the assumption that training data is a normal sequence. The learned feature representations can be biased by the presence of anomalies or noise in the training data. Thresholds are difficult to set, being very sensitive and exhaustive to find. Setting the ideal regularization is also hard, since it can easily underfit or overfit. If it has a good fit, it is possible to also generalize anomalies. There are specific problems associated with WDS anomaly detection. Leaks are easily confused with other events, such as unusual demands or maintenance activities. Besides that, it is hard to verify events detected by the models because sometimes there is no data about them. In synthetic data, it was verified that depends on the leak size. Bigger leaks and close to sensors are easily detected. The flow direction has also influence. Leaks can also be confused with demand if very close to the consumption point. It is possible to detect leaks close to a sensor with greater intensity but it also can detect other events occurring in other areas of the network. There is a necessity for other principles to overcome the identified anomalies.

VII. Future work

These architectures were identified has generalizing anomalies. To solve this drawback, a negative learning technique could be applied. This approach controls the compressing capacity of an autoencoder by optimizing the objectives of normal and abnormal data. But this would only work if data was labeled or partially labeled [24]. Variational Autoencoders (VAE) would work without the need of labels [2]. They have the advantage of not needing to set a threshold. The reconstruction probability is more objective than the reconstruction error. The reconstruction probability is a probabilistic measure that considers the variability of the distribution of variables.

It is possible to create different types of neural network layers and architectures under the autoencoder framework. It is an area that still requires research. To solve the problem of the normal sequence being polluted with anomalies or having noise, the idea of RPCA may be used in AEs to train more robust detection models [25]. This method learns a nonlinear subspace that captures most data points while allowing for some data to have arbitrary corruption. Another alternative is, instead of exclusively using loss based on the distances between series for training, sequences with labeled leaks could be used, and loss would be penalized if the model would reconstruct leaks wells. The worse the leak reconstructions, the lower would be the loss.

Since finding a good regularization is difficult a solution is not to regularize at all. Data is randomly partitioned into many equally sized parts, overfit each part with its autoencoder, and to use the maximum overall autoencoder reconstruction errors as the anomaly score [15].

With the insight obtained, these architectures can help to detect bursts. This could help to relax the already existent alarm on the WDS, reducing the number of false alarms. Also, the results obtained can help a domain expert, a civil engineer, to make decisions or to find interesting patterns. These models can select certain sequences that, under expert validation, can be annotated and later used for developing supervised methods.

References

[1] Kazeem Adedeji et al. "Leakage Detection and Estimation Algorithm for Loss Reduction in Water Piping Networks". In: *Water* 9 (Oct. 2017), pp. 1–21. DOI: 10.3390/w9100773.

[2] Jinwon An and S. Cho. "Variational Autoencoder based Anomaly Detection using Reconstruction Probability". In: 2015.

[3] Christoph Bergmeir and José Benítez. "On the use of cross-validation for time series predictor evaluation". In: *Information Sciences* 191 (May 2012), pp. 192–213. DOI: 10.1016/j.ins.2011.12.028.

[4] S.A. Boyer. *SCADA: Supervisory Control and Data Acquisition*. An Independent learning module from the Instrument Society of America. International Society of Automation, 2010. ISBN: 9781936007097. URL: https://books.google.pt/books?id=5q-qQAAACAAJ.

[5] Mikel Cañizo et al. "Multi-Head CNN-RNN for Multi-Time Series Anomaly Detection: An industrial case study". In: *Neurocomputing* (July 2019). DOI: 10.1016/j.neucom.2019.07.034.

[6] Jinghui Chen et al. "Outlier Detection with Autoencoder Ensembles". In: *SDM*. 2017.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[8] P. Hu et al. "A hybrid model based on CNN and Bi-LSTM for urban water demand prediction". In: *2019 IEEE Congress on Evolutionary Computation (CEC)*. June 2019, pp. 1088–1094. DOI: 10.1109/CEC.2019.8790060.

[9] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *CoRR* abs/1502.03167 (2015). arXiv: 1502.03167. URL: http://arxiv.org/abs/1502.03167.

[10] Amir Jalalkamali and Navid Jalalkamali. "Application of Hybrid Neural Modeling and Radial Basis Function Neural Network to Estimate Leakage Rate in Water Distribution Network". In: *World Applied Sciences Journal* 15 (Jan. 2011).

[11] T. Kieu, B. Yang, and C. S. Jensen. "Outlier Detection for Multidimensional Time Series Using Deep Neural Networks". In: *2018 19th IEEE International Conference on Mobile Data Management (MDM)*. June 2018, pp. 125–134. DOI: 10.1109/MDM.2018.00029.

[12] Tung Kieu et al. "Outlier Detection for Time Series with Recurrent Autoencoder Ensembles". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 2725–2732. DOI: 10.24963/ijcai.2019/378. URL: https://doi.org/10.24963/ijcai.2019/378.

[13] Tae-Young Kim and Sung-Bae Cho. "Predicting the Household Power Consumption Using CNN-LSTM Hybrid Networks: 19th International Conference, Madrid, Spain, November 21–23, 2018, Proceedings, Part I". In: Nov. 2018, pp. 481–490. ISBN: 978-3-030-03492-4. DOI: 10.1007/978-3-030-03493-1_50.

[14] Kwangsuk Lee et al. "Stacked Convolutional Bidirectional LSTM Recurrent Neural Network for Bearing Anomaly Detection in Rotating Machinery Diagnostics". In: July 2018. DOI: 10.1109/ICKII.2018.8569065.

[15] Boris Lorbeer and Max Botler. *Anomaly Detection With Partitioning Overfitting Autoencoder Ensembles*. 2020. arXiv: 2009.02755 [cs.LG].

[16] Pankaj Malhotra et al. "Long Short Term Memory Networks for Anomaly Detection in Time Series". In: Apr. 2015.

[17] Pankaj Malhotra et al. "LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection". In: *CoRR* abs/1607.00148 (2016). arXiv: 1607.00148. URL: http://arxiv.org/abs/1607.00148.

[18] Stephen R. Mounce and John Machell. "Burst detection using hydraulic data from water distribution systems with artificial neural networks". In: *Urban Water Journal* 3.1 (2006), pp. 21–31. DOI: 10.1080/15730620600578538. eprint: https://doi.org/10.1080/15730620600578538. URL: https://doi.org/10.1080/15730620600578538.

[19] Steve Mounce et al. "Pattern matching and associative artificial neural networks for water distribution system time series data analysis". In: *Journal of Hydroinformatics* 16 (May 2014), pp. 617–632. DOI: 10.2166/hydro.2013.057.

[20] Steve Mounce et al. "Sensor-fusion of hydraulic data for burst detection and location in a treated water distribution system". In: *Information Fusion* 4 (Sept. 2003), pp. 217–229. DOI: 10.1016/S1566-2535(03)00034-4.

[21] A. Nanduri and L. Sherry. "Anomaly detection in aircraft data using Recurrent Neural Networks (RNN)". In: *2016 Integrated Communications Navigation and Surveillance (ICNS)*. 2016, pp. 5C2-1-5C2–8.

[22] Raido Puust et al. "A review of methods for leakage management in pipe networks". In: *URBAN WATER JOURNAL* 7 (Feb. 2010), pp. 25–45. DOI: 10.1080/15730621003610878.

[23] H. Sak, Andrew Senior, and F. Beaufays. "Long short-term memory recurrent neural network architectures for large scale acoustic modeling". In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* (Jan. 2014), pp. 338–342.

[24] Maximilian Stubbemann, Tom Hanika, and Gerd Stumme. "Orometric Methods in Bounded Metric Data". In: *Advances in Intelligent Data Analysis XVIII - 18th International Symposium on Intelligent Data Analysis, IDA 2020, Konstanz, Germany, April 27-29, 2020, Proceedings*. Ed. by Michael R. Berthold, Ad Feelders, and Georg Krempl. Vol. 12080. Lecture Notes in Computer Science. Springer, 2020, pp. 496–508. DOI: 10.1007/978-3-030-44584-3_39. URL: https://doi.org/10.1007/978-3-030-44584-3%5C_39.

[25] Chong Zhou and Randy Paffenroth. "Anomaly Detection with Robust Deep Autoencoders". In: Aug. 2017, pp. 665–674. DOI: 10.1145/3097983.3098052.

[26] M. Zhou et al. "Leak Detection and Location Based on ISLMD and CNN in a Pipeline". In: *IEEE Access* 7 (2019), pp. 30457–30464. DOI: 10.1109/ACCESS.2019.2902711.