Investigating the Suitability of Concept Drift Detection for Detecting Leakages in Water Distribution Networks

Valerie Vaquet*[a], Fabian Hinder*[b] and Barbara Hammer[c]
Machine Learning Group, Bielefeld University, Germany

Keywords: Concept Drift Detection, Water Distribution Networks, Anomaly Detection, Leakage Detection.

Abstract: Leakages are a major risk in water distribution networks as they cause water loss and increase contamination risks. Leakage detection is a difficult task due to the complex dynamics of water distribution networks. In particular, small leakages are hard to detect. From a machine-learning perspective, leakages can be modeled as concept drift. Thus, a wide variety of drift detection schemes seems to be a suitable choice for detecting leakages. In this work, we explore the potential of model-loss-based and distribution-based drift detection methods to tackle leakage detection. We additionally discuss the issue of temporal dependencies in the data and propose a way to cope with it when applying distribution-based detection. We evaluate different methods systematically for leakages of different sizes and detection times. Additionally, we propose a first drift-detection-based technique for localizing leakages.

1 INTRODUCTION

Clean and safe drinking water is a scarce resource in many areas. Almost 80% of the world's population is classified as having high levels of threat in water security (Vörösmarty et al., 2010). This will aggravate in the future as due to climate change the already limited water resources will become more restricted (Rodell et al., 2018). Currently, across Europe, considerable amounts of drinking water are lost due to leakages in the system[1].

To ensure a reliable drinking water supply, there is a need for robust, safe, and efficient water distribution networks (WDNs). In addition to avoiding water losses, a crucial requirement is to ensure the quality of the drinking water. As leakages enable unwanted substances to enter the water system, monitoring the system for leakages is an efficient tool to avoid water loss and contamination (Eliades and Polycarpou, 2010; Lambert, 1994).

Due to complex network dynamics and changing demand patterns detecting leakages is a challenging task. This is aggravated by the fact that the available data is very limited. Usually, the precise network topology remains unknown or the documentation contains errors. As smart meter technologies are not widely distributed there is no real-time demand information (Cardell-Oliver and Carter-Turner, 2021). In realistic settings, this leaves a set of scarce pressure and possibly flow measurements.

Commonly, existing leakage detection methodologies rely on replicating the system of interest by hydraulic models and monitoring the discrepancies between observations and modeled values. While these approaches can provide reasonable detection when considering larger leakages, the approaches struggle when facing smaller leakages (Vrachimis et al., 2022). Besides, limited (real-time) information on the system is hindering the usage of these applications in real-world applications and the methodologies lack generalizability. Next to the hydraulic approaches, there are also a few machine learning (ML)-based approaches that implement a similar strategy.

In this work, we focus on the problem of leakage detection from the perspective of handling data streams containing temporal dependencies. More precisely, we formalize leakages as concept drift and the problem of leakage detection as drift detection. We aim to investigate the suitability of drift detection for reliable leakage detection, whereby we focus on leakage of all practically relevant sizes. Our approach is independent of the specific WDN and requires only

[a] https://orcid.org/0000-0001-7659-857X
[b] https://orcid.org/0000-0002-1199-4085
[c] https://orcid.org/0000-0002-0935-5591

[1] https://www.eureau.org/resources/publications/1460-eureau-data-report-2017-1/file
∗ authors contributed equally

Vaquet, V., Hinder, F. and Hammer, B.
Investigating the Suitability of Concept Drift Detection for Detecting Leakages in Water Distribution Networks.
DOI: 10.5220/0012361200003654
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2024)*, pages 296-303
ISBN: 978-989-758-684-2; ISSN: 2184-4313
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

real-time pressure measurements. Thus, it is more flexible and efficient than hydraulic simulation-based approaches.

This paper is structured as follows. First, we introduce WDNs and summarize the main specifics of this domain (section 2). Afterward, we briefly summarize the body of related work on leakage detection (section 3). In section 4, we define concept drift and cover model-loss-based and distribution-based drift detection. Before evaluating the suitability of these methodologies for leakage detection in section 6, we discuss the issue of temporal dependencies in the data collected from WDNs and propose ways to account for those (section 5). Finally, we conclude our paper in section 7.

2 WATER DISTRIBUTION NETWORKS

WDNs can be modeled as graphs consisting of nodes representing junctions and undirected edges representing pipes as the flow direction of the water is not pre-defined and can change over time due to changing demands in the network. As the systems are observed over time, hydraulic quantities like pressure and flow describe the network graph at each time step. They can be described by hydraulic formulas given that the network is known in great detail. Next to the exact pipe layout, different parameters like elevations, pipe diameters, and pipe roughness are required. Assuming sensors are installed in n nodes across the graph, for each time step t, measurements $x_t = [x_1^t \dots x_n^t]$ are collected where x_i^t is the value at node i.

Real-world measurements of the complete hydraulic state of WDNs are not available as it is not possible to measure the entire system. Even a precise topology alongside measurements in many positions is usually not available. When developing and evaluating methodologies monitoring WDNs one usually relies on simulated data. Given the key parameters of a system (layout, elevations, pipe information) alongside demand patterns realistic network states containing anomalies like leakages can be simulated using simulation tools like EPANET (Rossman, 2000).

As WDNs are part of the critical infrastructure and are required to work robustly and safely to ensure the health and well-being of the population, additional requirements are put on monitoring tools, especially those using AI technologies. Besides requirements concerning robustness, safety, and fairness as formulated in the European AI-Act (European Commission, 2021), some technical attributes of WDNs pose additional challenges to ML approaches. When working with WDNs, only limited knowledge about the pipe system is available. Usually, the exact properties of the pipes, e.g. their diameters and roughness, and the different elevation levels are unknown. Note that these are available for a few benchmark networks, and thus benchmark scenarios can be generated. However, when designing monitoring systems relying on this kind of information strongly limits the applicability in practical applications.

Besides limited information about the system setup, the system is also relatively opaque concerning the real-time dynamics. Due to installation costs and challenges regarding the power supply, the availability of pressure and flow sensors in WDNs is very limited yielding readings at a fraction of the nodes in the system. Data availability is even more limited for real-time demand measurements, as households are very rarely equipped with smart meters for drinking water due to costs and data privacy (Cardell-Oliver and Carter-Turner, 2021).

Another property of WDNs is the presence of cyclic patterns in demands, flows, and pressures. When working on ML approaches, one needs to account for the presence of temporal dependencies. Daily, weekly, and seasonal patterns as well as long-term developments, e.g. climate change or the COVID pandemic, increase the difficulty of leakage detection as especially smaller leakages might be lost in the signals.

3 RELATED WORK

The body of related work on leakage detection can be divided into methods relying on a hydraulic model and very few ML-based approaches. Hydraulic model-based methods generally aim to replicate the real-world system with a hydraulic model (Hu et al., 2018). Usually, the simulation results of the hydraulic model are then compared to the observations. An anomaly is reported if the residual of these methods is too large, which is determined either by a threshold (Romero-Ben et al., 2022), a CUSUM approach (Steffelbauer et al., 2022), or visual inspection (Marzola et al., 2022). All these methods share the downside that they require real-time demands and more information on the network topology than is usually available (Vrachimis et al., 2022). Besides, they lack generalizability across WDNs as the hydraulic model is specifically designed for one network and even needs adaptation if something changes within this particular network. While these hydraulic-based approaches yield good results considering large leakages they usually miss smaller ones (Vrachimis et al., 2022).

There are few ML-based approaches for leakage detection (Daniel et al., 2022; Laucelli et al., 2016; Romano et al., 2014). However, many are only evaluated on very small networks and lack realistic demands as input for the simulation data. Most of these approaches replace the hydraulic model with some ML model following the general idea of residual-based anomaly detection, for example by using a threshold (Daniel et al., 2022; Laucelli et al., 2016).

4 DETECTING CONCEPT DRIFT

Deploying ML-based systems in real-world scenarios, one needs to account for all kinds of changes and ensure that the models reliably work even if the observed environment changes. Thus, considerable research focuses on ML in the presence of changes in the data-generating process, which are called concept drift or drift for shorthand. To obtain a formal definition of drift, we first need to define a so-called drift process (Hinder et al., 2020; Hinder et al., 2023c):

Definition 1. Let $\mathcal{T} = [0,1]$ and $\mathcal{X} = \mathcal{R}^d$. A *drift process* $(P_T, \mathcal{D}_t)$ from the *time domain* $\mathcal{T}$ to the *data space* $\mathcal{X}$ is a probability measure P_T on $\mathcal{T}$ together with a Markov kernel $\mathcal{D}_t$ from $\mathcal{T}$ to $\mathcal{X}$, i.e. for all $t \in \mathcal{T}$ $\mathcal{D}_t$ is a probability measure on $\mathcal{X}$ and for all measurable $A \subset \mathcal{X}$ the map $t \mapsto \mathcal{D}_t(A)$ is measurable. We will just write $\mathcal{D}_t$ instead of $(P_T, \mathcal{D}_t)$ if this does not lead to confusion.

Based on this a definition of drift can be obtained:

Definition 2. Let $(P_T, \mathcal{D}_t)$ be a drift process. We say that $\mathcal{D}_t$ has *drift* iff

$$\mathbb{P}_{T,S\sim P_T}[\mathcal{D}_T \neq \mathcal{D}_S] = P_T^2(\{(t,s) \in \mathcal{T}^2 \mid \mathcal{D}_t \neq \mathcal{D}_s\}) > 0.$$

In many monitoring settings, the goal is to detect the drift by using *model-loss-based* or *distribution-based* approaches. While the latter directly investigates the observed data, model-loss-based approaches first train a model and then analyze its loss as a proxy for change in the data distribution. The rationale is that a drift event changes the data so that the model cannot approximate well anymore, causing a decline in the model loss. As argued by (Hinder et al., 2023a; Hinder et al., 2023b) the relation between model-loss and drift is rather loose – in case the model does not provide sufficient complexity to approximate the data distribution well (i) the drift might stay undetected as it is smoothed out by the model or in converse (ii) the model might change because of irrelevant changes, e.g. a change in the ratio of classes. Thus, from a theoretical point of view, one should rely on distribution-based drift detection. However, model-loss-based approaches like the residual-based strategy described in section 3, are also widely used in monitoring tasks. Therefore, we will investigate the suitability of both types of drift detection methods in this work.

4.1 Model-Loss-Based Drift Detection

Applying model-loss-based drift detection, there are two reasonable inference tasks a model can perform as a proxy for the drift detection: Either one performs a *forecasting* task where the goal is to predict the measurement of next time step x_{t+1} based on the sensor measurements collected up to time t, or one performs an *interpolation* task where the goal is to predict one sensor by the measurements of all other sensors, i.e. for each node position i, a model $f_i : \mathcal{R}^{n-1} \to \mathcal{R}$, $f_i(x^t_{\setminus i}) = \hat{x}^t_i$ is trained, where $x^t_{\setminus i}$ means we take all measurements but that of node i at time t. The latter strategy has been employed as a virtual sensor imputation strategy in case of sensor faults. Even very simple ML models could successfully perform the interpolation task (Vaquet et al., 2022). As we observed worse results for forecasting in preliminary experiments, we only cover interpolation in this work.

4.2 Distribution-Based Drift Detection

Most distribution-based approaches follow the strategy of *comparing two samples* (Hinder et al., 2023c). This can be done by *statistical testing*, e.g. by using the *Kolmogorov-Smirnow (KS) test* (Kolomogorov, 1933) feature-wise or the *kernel two-sample test* which relies on the maximum mean discrepancy (MMD) and uses a kernel matrix as a descriptor (Gretton et al., 2006). Another option is using a *virtual classifier* discriminating between the two windows. In case it performs better than guessing, the distributions of the windows differ, i.e. a drift occurred. We will consider the D3 detection scheme (Gözüaçık et al., 2019) in our experiments.

We additionally consider a *block-based* detection scheme searching directly for a dependency of data and time which was identified to be an equivalent description of drift by (Hinder et al., 2020). This task can be performed by a standard independence test; in this work, we will make use of the HSIC-test (Gretton et al., 2007) which is another kernel-based method.

As discussed in section 2, different kinds of daily, weekly, and seasonal patterns have to be expected. These patterns introduce certain temporal dependencies to the data. As already discussed, these patterns might increase the difficulty of detecting leakages. Considering this from a theoretical viewpoint, this problem can be summarized by the need to account for the temporal dependencies when perform-

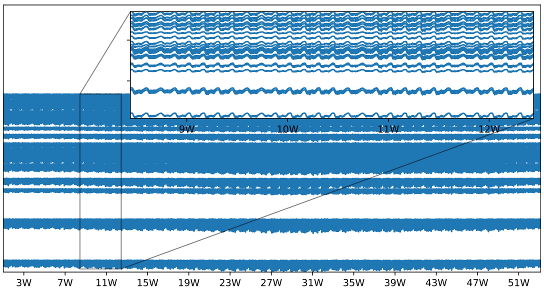

Figure 1: Sensor data for one year (no leakage).

ing drift detection. Thus, in the next section, we will analyze the temporal patterns in the data.

5 TEMPORAL DEPENDENCIES IN THE DATA

We already raised the issue of temporal dependencies in data collected from WDNs. In this section, we will analyze the dataset which we will use in our experiments later on. For this purpose, we will first briefly introduce the dataset and provide our analysis.

5.1 L-Town Benchmark Data

In this work, we will consider the L-Town network since it is relatively complex in comparison to other benchmarks and one year of realistic real-time demands are available for this system allowing us to simulate realistic data for our experimental evaluation. The L-Town network resembles parts of the old town of Limassol, Cyprus. In our experiments, we consider area A consisting of 661 nodes and 764 edges with 29 optimally placed pressure sensors (Vrachimis et al., 2022). We run simulations with four different leakage sizes ranging from 7mm to 19mm at all pipes using the ATMN package which builds on EPANET. Each scenario contains data for 364 days with a measuring frequency of 15 minutes. We always consider one leakage per scenario which starts at some point of the scenario and stays present until the scenario ends.

5.2 Analysis

Analyzing the data, as expected we observe daily, weekly, and seasonal patterns. As visualized in fig. 1, the pressure follows a clear weekly pattern as can be seen in the zoom-in subplot. To control those dependencies we perform two analysis strategies: 1) subtracting the “standard week”, 2) subtracting the values of the previous week.

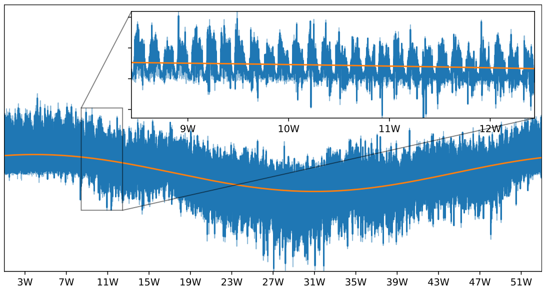

Figure 2: Sensor residuals after subtracting the standard week (no leakage). The orange line marks the mean trend across all sensors.

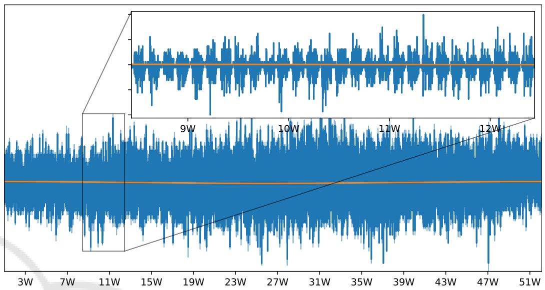

Figure 3: Sensor residuals after subtracting the value of last week (no leakage). The orange line marks the mean trend across all sensors.

By subtracting the standard week from the original signals we obtain the signals shown in fig. 2. The plot shows one example sensor reading (blue line) as well as the minimal and maximal sensor reading at each given point in time as the shaded area. As can be seen, the feature runs across the entire range implying very strong fluctuations. Furthermore, as can be seen in the zoomed-in plot there is a change in fluctuation that follows a daily pattern. We also added a trend line (orange) which follows a cosine shape. This is a plausible finding as we expect a cyclic pattern across several years that correlates with the seasons. However, this pattern may render change detection schemes useless as it induces changes that are not caused by leaks.

As an alternative, we considered subtracting the value of the last week rather than a standard week. The results are illustrated in fig. 3. Due to the small variance in the computation of the standard week, this is already a good proxy for the standard week. However, it is better suited to cope with long-term changes as can be seen from the trend line. Furthermore, we again observe strong oscillations whose intensities follow a daily pattern. We will find that this strategy is quite efficient in section 6.1.

From both analyses, we expect that we can easily cope with the periodic patterns if we only compare the data on a by-week basis. This is because there is a noticeable difference between the values of weekends and weekdays so that day-wise is too short to resolve this dependency. Furthermore, longer periods will be strongly affected by the seasonal trends. We will further discuss those ideas in the next section.

5.3 Coping with Temporal Dependencies

We observed substantial temporal patterns in the data which we need to account for when utilizing drift detection schemes for the task of detecting leakages. For model-loss-based drift detection approaches we assume that the models can generalize well. Thus, in this setting, no additional actions need to be taken. In contrast, when using distribution-based schemes, we need to carefully incorporate our knowledge of the different temporal cycles in the data to successfully detect leakages.

In preliminary experiments, we used a preprocessing technique, which subtracted a standard week to eliminate cycles in the data. However, this strategy assumes that we can model this standard week successfully which requires some leakage-free historical data. Since we aim to develop a methodology that requires as little information as possible to generalize to new networks, we additionally experimented with choosing the window sizes such that the detection schemes do not suffer from seasonalities. Here, as discussed before, our idea is to eliminate the cyclic patterns by choosing exactly one week per window. Thereby daily and weekly patterns are eliminated while the windows are still small enough to not be affected by long-term dependencies. Since this strategy resulted in better results while requiring no additional information, we will use this option in our experimental evaluation instead of performing a preprocessing step subtracting the standard or the previously observed week.

6 EXPERIMENTS

For all our experiments[2], we use the data benchmark which we described in section 5.1.

6.1 Model-Loss-Based Drift Detection

To evaluate the model-loss-based detection schemes we rely on different regression models: *k*NN, polynomial ridge regression, random forests, and linear ridge regression. RBF-ridge and RBF-/Poly-/Linear-SVR were considered but discarded after initial considerations due to weak performances in the regression task. In our experiments, we first analyze the models' performance on the interpolation task, and their generalization capabilities to out-of-sample examples, e.g. to scenarios containing leakages. In the second step, we analyze how well the schemes are suited for detecting leakages. To do so we check the underlying assumption that the model would perform better on the original training data (without leakage) compared to the leaky data. For this to facilitate a useful strategy we need to be able to define a threshold θ such that $\mathrm{MSE}(x_t) > \theta$ indicates a leakage at time t and vice versa. Considering this as a classification problem with the classes "no leakage" and "leakage" we can apply the ROC-AUC score to evaluate the performance of our models. To be more resilient to slowly growing leakages we do not consider model updates. Thus, we end up with the following procedure:

1. Select one fold. Extract two consecutive weeks from the baseline dataset
2. Train the interpolation model on the data
3. Compute the errors of the model for the remaining year for each data point E_0
4. Compute the errors of the model for the entire year for each leakage location and size E_1
5. Compute the detection performance for this fold $\mathrm{ROC\text{-}AUC}([0,\ldots,0\ ,\ 1,\ldots,1], E_0 + E_1)$

Recall that the ROC-AUC measures how well the obtained scores separate the leaky and non-leaky setups. The score is 1 if the largest error without leakage is smaller than the smallest error with leakage, it is 0.5 if the assignment is random. Thus, the ROC-AUC provides a scale-invariant upper bound on the performance of every concrete threshold. It is not affected by class imbalance.

The results of our experiment evaluating the generalization ability are summarized in fig. 4. The observed errors increase with increasing leakage diameters and the models generalize to small leakages. In this setting, we find that the simple linear models perform much better than the *k*NN and the random forest. This aligns with findings published in (Vaquet et al., 2022).

The evaluation of the detection experiments is summarized in fig. 5. We observe reasonable ROC-AUC scores for leakage sizes of about 15mm-19mm. However, small leakages pose a difficult problem for model-loss-based drift detection schemes: the scores are only marginally above random guessing.

In summary, we find that model-loss-based detection schemes are only suitable for detecting large

[2]The experimental code and hyperparameters are available at https://github.com/FabianHinder/Drift-and-Water

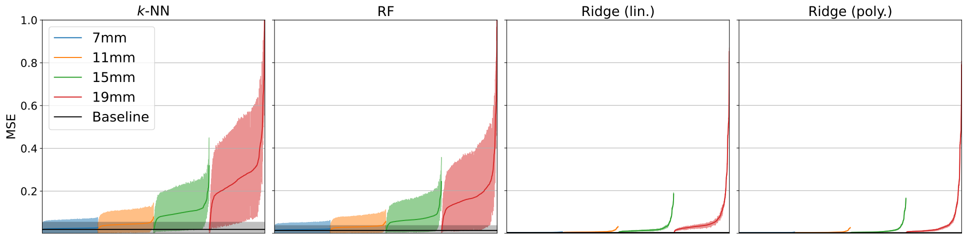

Figure 4: Mean squared error of the interpolation for different leakage sizes. Sorted (x-axis) ascending for each leakage size and mean error for clarity. Setting without leakage is reported as baselines. Line marks mean value, shaded area is minimal value to mean+standard deviation.

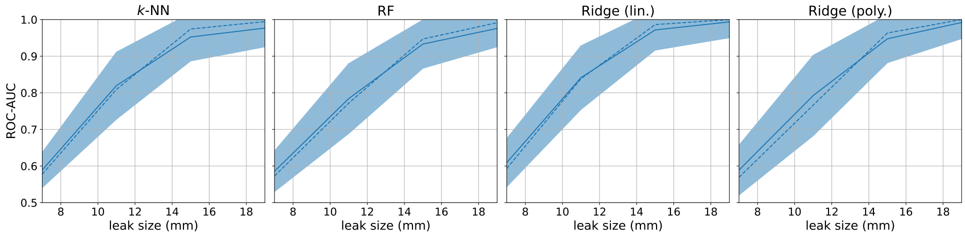

Figure 5: ROC-AUC for the model-loss-based drift detection for increasing leakage sizes. Interpolation task. Solid line marks mean value, shaded area mean±standard deviation, dashed line median.

leakages since the models generalize too well to out-of-distribution samples to reliably detect small leakages. This aligns with both results of other works and the theory we briefly discussed in section 4.

6.2 Distribution-Based Drift Detection

Concerning distribution-based drift detection (see section 4.2), as argued before, choosing a suitable window size is crucial for the successful usage of distribution-based drift detection schemes. Analyzing the system we decided to rely on two windows of one week each to eliminate the temporal patterns from the data. In addition to the choice of window size, the position of the split point, i.e. the border of the two considered windows, affects the performance of the detection scheme. The larger the displacement from the actual leakage onset time, the harder the detection will be. However, as it is desirable to detect leakages as soon as possible, it would be desirable to detect leakages even if the displacement is still large, e.g. if the split point still lies before the leakage occurred.

Fig. 6 summarizes the results of the distribution-based detection schemes with the different models. Assuming the split point lies exactly on the leakage, we report much better detection results than for the model-loss-based detection strategies. We report smaller scores for DAWIDD which is to be expected in this case as the model is not benefiting from our window size choice due to its block-based nature and is thus affected by the temporal dependencies in the oscillations. While we obtain smaller scores for the statistical test-based methods for small leakages, the virtual classifier-based methods reliably detect leakages of a diameter of 7mm.

As assumed, concerning the position of the split point, we observe a decline in the score with increasing displacement. However, even for a displacement of 4 days, we obtain better scores than for the best model-based detection schemes. For a displacement of 6 days, we still get reasonable scores for large leakages when using the D3 detection scheme with a linear model. Thus, detecting large leakages can be realized very fast, while for smaller leakages it takes a little more time to obtain a reliable warning.

These findings can be confirmed when analyzing fig. 7. One can see that apart from the smallest leakage size, with a displacement of 4 days, i.e. 3 days after the leakage occurred, most detection schemes yield a reasonable score. Using the linear version of D3 even the smallest considered leakages can be detected with a delay of 5 days (displacement of 2 days).

In conclusion, we found that distribution-based detection schemes outperform model-based detection across all leakage sizes if the window size is chosen suitably. Large leakages can be detected with a reasonably small delay. In practical applications implementing a warning mechanism early on could be realized to react early on.

6.3 Leakage Localization

Besides the task of leakage detection, there is also the more specific task of leakage localization, i.e. deter-

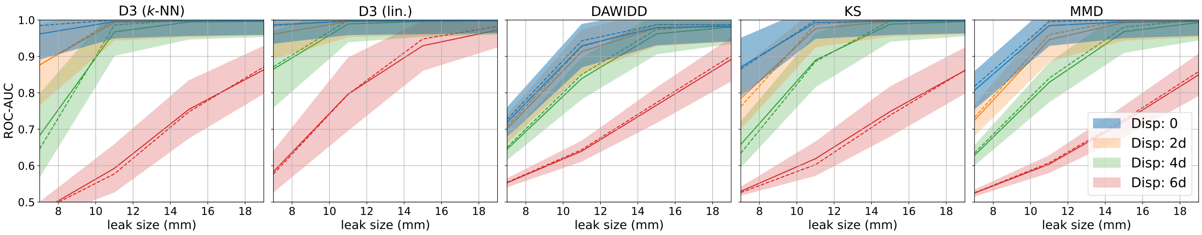

Figure 6: Performance of unsupervised drift detectors for different displacements (discrepancy between split point (assumed time-point of drift) and actual drift). Solid line marks mean value, shaded area mean±standard deviation, dashed line median.

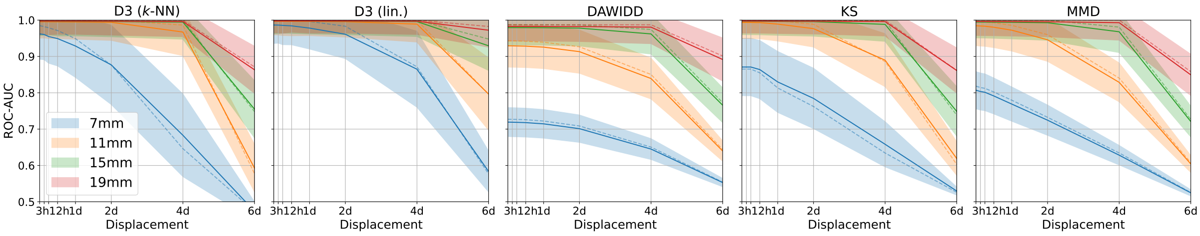

Figure 7: Performance of unsupervised drift detectors for different leakage sizes. Solid line marks mean value, shaded area mean±standard deviation, dashed line median.

mining the pipe where the leakage occurs (Vrachimis et al., 2022). This task is usually considered much harder and is commonly approached by formulating an inverse problem, i.e. evaluating the plausibility of different locations (Li et al., 2022; Daniel et al., 2022; Wang et al., 2022; Marzola et al., 2022). This again requires a lot of data usually not available to us. As our drift detection approach performed quite well on the detection task we consider the possibility to extend the methodology to leakage localization. Here the idea is quite simple: the closer a sensor is to the leakage's actual position, the stronger the influence and thus the drift, leading to a particularly small p-value for that feature. As only the Kolmogorov-Smirnov test operates feature-wise we consider this scheme using the same same setup as before. We return the sensor node that has the smallest p-value, i.e. is considered to be particularly drifting by the test.

In the following let S be all sensor nodes, s^* be the selected sensor node, and v be the node where there leakage actually occurred, even if it is not the sensor node. Furthermore, d denotes the graph distance in the WDN, i.e. $d(a,b)$ is the length of the shortest path connecting a and b. We make use of three metrics: distance between selected and actual node (Dist.; $d(s^*,v)$), number of sensor nodes closer to the actual node (#Cls.; $|\{s \in S \mid d(s,v) < d(s^*,v)\}|$), and relative distance between actual node, selected and optimal node (rel.D.; $d(s^*,v)/\min_{s\in S} d(s,v)$) which is normalized in contrast to the simple distance and smooth in contrast to the closer node metric.

The results are shown in table 1. They are quite promising. We observe that the precision is decreasing for smaller leakage sizes, which is to be expected considering the results from the last experiment.

Table 1: Results of leakage localization.

size (mm)	Dist. $\mu\pm\sigma$	rel.D. $\mu\pm\sigma$	#Cls. $\mu\pm\sigma$
7	10.1±13.1	2.6±4.9	3.3±7.0
11	5.5±4.7	1.3±1.4	0.6±2.0
15	5.1±3.9	1.2±1.2	0.5±1.5
19	5.0±3.6	1.2±1.1	0.4±1.4

7 CONCLUSION

In this work, we investigated the suitability of model-loss-based and distribution-based drift detection methods. Combining distribution-based detection with knowledge of WDNs, we provide detection schemes that successfully detect leakages of all sizes with reasonable detection delays. Analyzing model-loss-based techniques that are widely implemented in the water domain, we confirmed theoretical results that raise the issue of the loose connection between model loss and drift.

We assume that our work is not limited to WDNs but can also be realized for anomaly detection in other critical infrastructure systems like gas or electrical grids. In practical applications a further analysis of the leakages is necessary – solely detecting leakages is not sufficient to take appropriate actions. We proposed a first localization strategy that is based directly on detection efforts. Considering these follow-up tasks in more detail through the lens of concept drift both practically and theoretically is an interesting path for further research.

ACKNOWLEDGEMENTS

We gratefully acknowledge funding from the European Research Council (ERC) under the ERC Synergy Grant Water-Futures (Grant agreement No. 951424).

REFERENCES

European Commission (2021). Artificial Intelligence Act.

Cardell-Oliver, R. and Carter-Turner, H. (2021). Activity-aware privacy protection for smart water meters. In *8th ACM BuildSys*, BuildSys '21, page 31–40. Association for Computing Machinery.

Daniel, I., Pesantez, J., Letzgus, S., Khaksar Fasaee, M. A., Alghamdi, F., Berglund, E., Mahinthakumar, G., and Cominola, A. (2022). A Sequential Pressure-Based Algorithm for Data-Driven Leakage Identification and Model-Based Localization in Water Distribution Networks. *J. Water Resour. Plan. Manag.*, 148.

Eliades, D. G. and Polycarpou, M. M. (2010). A Fault Diagnosis and Security Framework for Water Systems. *IEEE Transactions on Control Systems Technology*, 18(6):1254–1265.

Gözüaçık, Ö., Büyükçakır, A., Bonab, H., and Can, F. (2019). Unsupervised concept drift detection with a discriminative classifier. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2365–2368.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. J. (2006). A kernel method for the two-sample-problem. In *NIPS 2006*, pages 513–520.

Gretton, A., Fukumizu, K., Teo, C. H., Song, L., Schölkopf, B., and Smola, A. J. (2007). A kernel statistical test of independence. In *NIPS 2007*, pages 585–592.

Hinder, F., Artelt, A., and Hammer, B. (2020). Towards non-parametric drift detection via dynamic adapting window independence drift detection (DAWIDD). In *ICML 2020*, volume 119, pages 4249–4259. PMLR.

Hinder, F., Vaquet, V., Brinkrolf, J., and Hammer, B. (2023a). On the Change of Decision Boundary and Loss in Learning with Concept Drift. In *IDA XXI*, volume 13876, pages 182–194. Springer Nature Switzerland, Cham.

Hinder, F., Vaquet, V., Brinkrolf, J., and Hammer, B. (2023b). On the Hardness and Necessity of Supervised Concept Drift Detection:. In *12th ICPRAM*, pages 164–175, Lisbon, Portugal. SCITEPRESS.

Hinder, F., Vaquet, V., and Hammer, B. (2023c). One or two things we know about concept drift – a survey on monitoring evolving environments.

Hu, C., Li, M., Zeng, D., and Guo, S. (2018). A survey on sensor placement for contamination detection in water distribution systems. *Wireless Networks*, 24(2):647–661.

Kolomogorov, A. (1933). Sulla determinazione empirica di una legge didistribuzione. *Giorn Dell'inst Ital Degli Att*, 4:89–91.

Lambert, A. (1994). Accounting for Losses: The Bursts and Background Concept. *Water and Environment Journal*, 8(2):205–214.

Laucelli, D., Romano, M., Savić, D., and Giustolisi, O. (2016). Detecting anomalies in water distribution networks using EPR modelling paradigm. *Journal of Hydroinformatics*, 18(3):409–427.

Li, Z., Wang, J., Yan, H., Li, S., Tao, T., and Xin, K. (2022). Fast Detection and Localization of Multiple Leaks in Water Distribution Network Jointly Driven by Simulation and Machine Learning. *J. Water Resour. Plan. Manag.*, 148(9).

Marzola, I., Mazzoni, F., Alvisi, S., and Franchini, M. (2022). Leakage Detection and Localization in a Water Distribution Network through Comparison of Observed and Simulated Pressure Data. *J. Water Resour. Plan. Manag.*, 148(1):04021096.

Rodell, M., Famiglietti, J. S., Wiese, D. N., Reager, J., Beaudoing, H. K., Landerer, F. W., and Lo, M.-H. (2018). Emerging trends in global freshwater availability. *Nature*, 557(7707):651–659.

Romano, M., Kapelan, Z., and Savić, D. A. (2014). Automated Detection of Pipe Bursts and Other Events in Water Distribution Systems. *J. Water Resour. Plan. Manag.*, 140(4):457–467.

Romero-Ben, L., Alves, D., Blesa, J., Cembrano, G., Puig, V., and Duviella, E. (2022). Leak Localization in Water Distribution Networks Using Data-Driven and Model-Based Approaches. *J. Water Resour. Plan. Manag.*, 148(5).

Rossman, L. A. (2000). EPANET 2: users manual. US Environmental Protection Agency. Office of Research and Development.

Steffelbauer, D. B., Deuerlein, J., Gilbert, D., Abraham, E., and Piller, O. (2022). Pressure-Leak Duality for Leak Detection and Localization in Water Distribution Systems. *J. Water Resour. Plan. Manag.*, 148(3).

Vaquet, V., Artelt, A., Brinkrolf, J., and Hammer, B. (2022). Taking Care of Our Drinking Water: Dealing with Sensor Faults in Water Distribution Networks. In *ICANN 2022*, volume 13530, pages 682–693. Springer Nature Switzerland, Cham.

Vörösmarty, C. J., McIntyre, P. B., Gessner, M. O., Dudgeon, D., Prusevich, A., Green, P., Glidden, S., Bunn, S. E., Sullivan, C. A., Liermann, C. R., et al. (2010). Global threats to human water security and river biodiversity. *nature*, 467(7315):555–561.

Vrachimis, S. G., Eliades, D. G., Taormina, R., Kapelan, Z., Ostfeld, A., Liu, S., Kyriakou, M., Pavlou, P., Qiu, M., and Polycarpou, M. M. (2022). Battle of the Leakage Detection and Isolation Methods. *Journal of Water Resources Planning and Management*, 148(12).

Wang, X., Li, J., Liu, S., Yu, X., and Ma, Z. (2022). Multiple Leakage Detection and Isolation in District Metering Areas Using a Multistage Approach. *J. Water Resour. Plan. Manag.*, 148(6).