LeakDB: A benchmark dataset for leakage diagnosis in water distribution networks

Stelios G. Vrachimis *[1], **Marios S. Kyriakou**[2], **Demetrios G. Eliades**[3], **and Marios M. Polycarpou**[4]

[1,2,3,4]KIOS Research and Innovation Center of Excellence, Dept. of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus,

* *vrachimis.stelios@ucy.ac.cy*

ABSTRACT

The increase of streaming data from water utilities is enabling the development of real-time anomaly and fault detection algorithms that can detect events, such as pipe bursts and leakages. Currently, there is not a widely accessible dataset of real or realistic leakage scenarios, which could be used as a common benchmark to compare different algorithms, as well as to support research reproducibility. In this work we propose the design of a realistic leakage dataset, the Leakage Diagnosis Benchmark (LeakDB). The dataset is comprised of a large number of artificially created but realistic leakage scenarios, on different water distribution networks, under varying conditions. Additionally, a scoring algorithm was developed in MATLAB to evaluate the results of different algorithms using various metrics. The usage of the LeakDB dataset, is demonstrated by scoring four detection algorithms. The dataset is stored on an open research data repository, and will be updated in the future with new simulation scenarios. The source code of the toolkit that generates the leakage benchmark dataset, as well as the detection algorithms used, are released as open source.

Keywords: Benchmark, Dataset, Leakage Diagnosis, Water Distribution Networks

1 Introduction

Developments in hydraulic sensor technologies and on-line data acquisition systems enable water companies to deploy a larger number of pressure and flow sensing devices within their systems. As a result, a large volume of streaming, time-series data is being collected, representing the state of hydraulic dynamics. These data are mainly used for regulatory reporting by water operators. In addition to reporting, these real-time data can be used for detecting events, such as pipe bursts and leakages. This is typically achieved by visual analysis and limit checking. In the literature, advanced anomaly and fault detection algorithms have been developed in the previous years, which are potentially more capable of distinguishing between actual events and false alarms, typically by comparing sensor measurements with some model. In practice, these algorithms need to be trained and evaluated on large volumes of data. However, researchers developing leakage detection and diagnosis algorithms, may have limited access to real datasets originating from industrial partners, as these may be considered as confidential. As a result, it is difficult for researchers to evaluate and compare their fault diagnosis methodologies on realistic water distribution systems.

Currently, there is not a widely accessible dataset of real or realistic leakage scenarios, which could be used as a common benchmark, as well as for research reproducibility. Such a dataset should contain

multiple scenarios and networks under varying conditions, to provide an objective assessment of the fault diagnosis algorithms. Benchmarks have been extensively used in multiple applications, such as image/video processing [1], body activity recognition [2], anomaly detection in times-series [3], and many more which are available in public repositories, such as the UCI Machine Learning Repository [1] and the Penn Machine Learning Benchmark [2]. In the water distribution systems analysis community, various EPANET network benchmarks have been presented in the past [4], as well as data provided in the context of conference competitions, e.g., for detecting cyber-physical attacks (BATtle of the Attack Detection ALgorithms - BATADAL) .

In this work we propose the *Leakage Diagnosis Benchmark* (LeakDB) dataset to promote research and evaluation of leakage detection and isolation algorithms. The requirements of the benchmark are analyzed based on systematic approaches for constructing benchmark datasets, e.g., as in [5]. The benchmark is comprised of a large number of realistic leakage scenarios which occur randomly at different water distribution benchmark networks, of different size and topology. For each benchmark network and for each leakage scenario, the leakage parameters (e.g., number of leaks, locations, size), the structural parameters (e.g., length, pipe roughness) and realistic consumer pressure-driven demands are varied. The dataset is comprised of all leakage scenario parameters, hydraulic dynamics (flows, pressures), node demands and the network model. The benchmark is described in Section 2.

Additionally, a scoring algorithm is provided to evaluate the results of the different algorithms using various metrics based on the confusion matrix, such as accuracy, precision, recall, F1-score, as well as the detection delay. The evaluation metrics used are explained in Section 3. To demonstrate the application of these metrics, a baseline detection algorithm developed by the authors is examined, which is based the concept of Minimum Night Flows (MNF) [6]. The evaluation of this algorithm is provided in Section 4. Finally, discussion and future work on this benchmark is provided in Section 5. The dataset is stored in an open research data repository given in Section 7. The source code of the toolkit that generates the leakage benchmark dataset, as well as the detection algorithm based on MNF are released under the EU Public License.

2 Description of Benchmark

The Leakage Diagnosis Benchmark (LeakDB) was created using the Water Network Tool for Resilience (WNTR), which is an open source Python package designed to help water utilities investigate resilience of water distribution systems to hazards [7]. The variations in different scenarios of the dataset are described below.

The networks in the dataset were carefully selected to have multiple varying characteristics in terms of size, topology and types of elements they contain. Varying the network size, i.e. the number of nodes and links, demonstrates the scalability of the algorithm. Networks with a different number of tanks, reveal the ability of detection algorithms to deal with dynamic states. Topology is also considered, specifically the complexity that arises in leakage diagnosis when the networks contain loops. This is quantified by calculating the circuit rank of the network, which is then normalized by the circuit rank of the same network if it was fully connected. The resulting metric is defined as the *Loop Ratio*, which has a value of zero when there are no loops in the network, while it is equal to one in the case of

[1] `http://archive.ics.uci.edu/ml/`

[2] `https://github.com/EpistasisLab/penn-ml-benchmarks`

a fully connected network graph. A table with the benchmark networks used and their characteristics is provided within the LeakDB.

The models of the networks used in the dataset are provided in the form of an EPANET 'INP' file. In different scenarios of the LeakDB these networks vary in terms of model parameters, thus introducing modeling uncertainty. These parameters are the pipe length, diameter and roughness, pump curves and settings and tank dimensions, which were varied in a range between $\pm\mu\%$ of their original value. In this work, this range was set to $\mu = 25\%$. The uncertainties are considered bounded and these bounds are provided in the dataset in the form of a percentage with respect to the provided model. Topological uncertainty is introduced in the form of unknown valve settings, i.e. at each scenario the initial status of n pipes (closed or opened) is randomly varied in respect to the provided model. The modified network model for each scenario is also provided in the dataset.

The loading conditions of the network can greatly affect the performance of algorithms, so different demand scenarios at the nodes of the network are considered. The demands are artificially created based on historical real-data from water utilities, which were decomposed into three signal components: (a) weekly periodic component, (b) yearly seasonal changes component and (c) random component [6]. The weekly periodic component describes the fluctuation of demand signals throughout one week, and depends on various social and economic characteristics of the consumers. The yearly seasonal changes component describes the variation in water consumption as a result of seasonality within a year. The random component considers the high frequency variations, which may be due to unpredictable consumer demands, transients, repairs and other network activities. All components are normalized around a zero mean in the range of $[-1, 1]$. Long-term trends are not considered in demand generation, as each scenario spans one year.

The creation of unique demand signals is based on the extracted components from real-data and a multiplicative formulation with respect to the signal components [8]. The two basic periodic components are approximated using Fourier Series (FS), as shown in Figure 1. For the weekly periodic component (a), a 20^{th} order FS approximation is used with period of one week. For the yearly seasonal changes component (b), a 3^{rd} order FS approximation is used with a period of one year. To create unique demand patterns that also resemble the original real patterns, the calculated FS coefficients are randomized in a range of $\pm\delta\%$ of their original value, where in this work $\delta = 10\%$. The resulting weekly periodic component and yearly seasonal component for each node are indicated by C_w and C_y respectively. The unique random component C_r is created by generating variables with a normal distribution around a zero mean and a standard deviation equal to ϵ, where in this work $\epsilon = 0.33$. Additionally, a base demand β is calculated by randomizing around an appropriate base demand value specific to each network. Finally, the unique demand pattern d for each node is generated by multiplying all components as follows: $d = \beta(C_w + 1)(C_y + 1)(C_r + 1)$. Note that the simulated demands in each scenario may be different from the demand pattern, due to the use of pressure-driven solver. All scenarios ensure that the network operates normally, given minimum pressure requirements.

Leakages are considered the only type of fault that can occur in the scenarios of the LeakDB dataset. These are simulated at network nodes and at each scenario the location, start time, end time and number of leakages is selected at random. The leakage magnitude varies due to the assignment of a random leakage hole diameter ϕ, which in this work was varied such that $\phi \in [2\text{cm} - 20\text{cm}]$, as well as due to the pressure at the leakage location. Each leakage is also assigned a time profile, categorizing them as *abrupt* leakages, or *incipient* leakages. Incipient leakages increase gradually

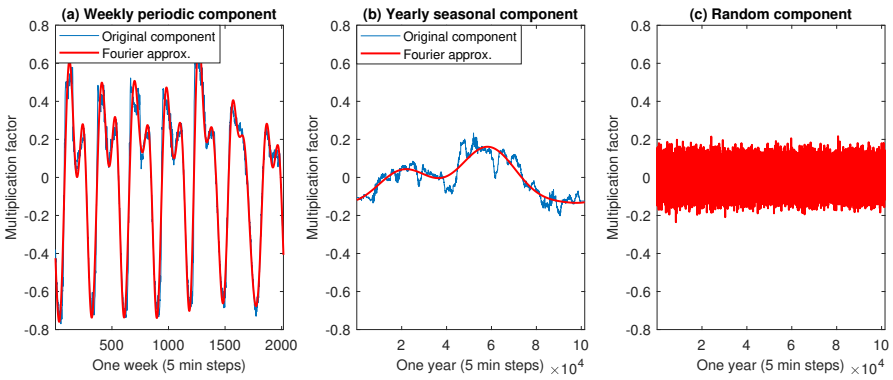

Figure 1: Demand components for demand time series construction

and are harder to detect by many algorithms that use past behavior to define anomalies in data, as they consider a new behavior can be anomalous at first but ceases to be anomalous if it persists; i.e. a new normal pattern is established. All information regarding the leakages occurring at each scenario is included in the dataset.

The LeakDB dataset divides the simulated scenarios of each network into different directories, so that network specific algorithm evaluations can be performed. Each *Network Directory* contains n_{sc} number of scenarios (e.g., 500 scenarios), with different number of leakage events (or no events). In each *Scenario Directory*, the modified network used is provided in EPANET 'INP' format. Additionally, for each scenario the node demands, node pressures, link flows and leak flow and info are provided in 'CSV' data files. Each data file has the same structure, containing the "Timestamp" column (`YYYY-MM-DD hh:mm:ss`) and a "Value" column (real numbers). The structure of the LeakDB and data file format are illustrated in Figure 2.

The LeakDB dataset is accompanied by *Label* files, indicating faults in binary notation. A Label file in each Network Directory, tags individual scenarios indicating if they contain a fault. Within each Scenario Directory, a Label file tags each individual time step of the scenario in which a fault exists. In the folder containing leakage information of each scenario, the start and end time of each individual fault, the leak hole diameter and leak type are provided.

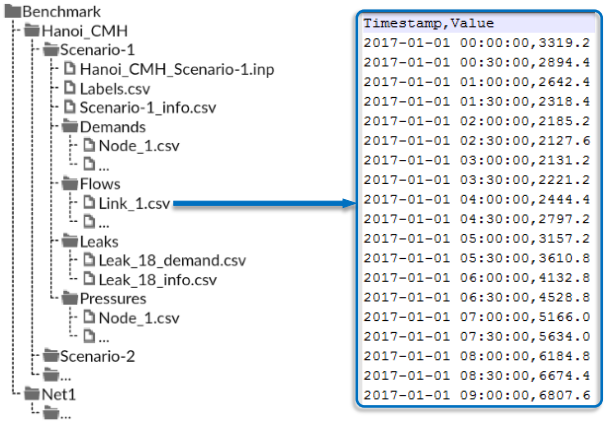

Figure 2: Structure of the LeakDB dataset and an example data file

3 Scoring Real-Time Leakage Detectors

The ideal leakage detection algorithm will have the following characteristics [3]: (C1) Detects all leakages present in the streaming data; (C2) Triggers no false alarms (no false positives); (C3) Detects leakages as soon as possible, ideally before the leakage becomes visible to a human; (C4) Works with real time data (no look ahead); (C5) Is fully automated across all datasets (parameter learning should be performed online). While characteristics C4 and C5 can either apply or not apply to a given algorithm, characteristics C1-C3 must be evaluated for each algorithm.

3.1 Classification ability score

In terms of classification accuracy, the scoring algorithm uses the data labels provided with the LeakDB to calculate the standard classification metrics of the confusion matrix: the true positive (TP), false positive (FP), true negative (TN), and false negative (FN). These are calculated for each time step of each scenario as follows:

$$TP = \sum_{i=1}^{n_{sc}} \sum_{k} TP_i(k), \tag{1}$$

with $TP_i(k)$ being a true positive in the *i*-th scenario on the *k*-th time step and n_{sc} is the number of scenarios considered. The rest of the elements of the confusion matrix TN, FP, FN are calculated in the same manner. Widely used metrics such as precision, recall, specificity and accuracy, can be computed based on the confusion matrix. Using these metrics, the ability of the algorithm to detect leakages in conditions where a leakage exists (C1) is quantified using the TP Rate (or recall) for all time steps in all scenarios, defined as:

$$\mathrm{R_{TP}} = TP/\left(TP + FN\right). \tag{2}$$

Additionally, the ability of the algorithm to avoid false alarms during the absence of leakages (C2) is quantified using the TN Rate (or specificity) for all time steps in all scenarios, defined as:

$$\mathrm{R_{TN}} = TN/\left(FP + TN\right). \tag{3}$$

In addition to the above, the F-measure (F1-score) is considered, which balances precision and recall of classifiers on each class by calculating their harmonic mean. The F-measure is a popular metric in datasets which are characterized by class imbalances. Class imbalance occurs when the total number of a class of data (positive) is significantly less than the total number of another class of data (negative). This problem is common in practice and can be observed in various disciplines including anomaly detection. The metric can be calculated separately by considering the confusion matrix computed for each scenario, or, by computing the total number of TP, FP and FN across all scenarios; the latter was shown to be unbiased in the case of class imbalances [9]. This is calculated as :

$$F_{tp,fp} = 2TP/\left(2TP + FP + FN\right). \tag{4}$$

3.2 Early detection score

The classification scores cannot evaluate effectively real-time detection algorithms as they do not incorporate time and do not reward early detection. Early detection (C3) in this work is evaluated

using the *early detection score* which rewards detections that are close to the start time of a fault t_f, while the reward declines as the detection moves further away [3]. The detection time t_d is defined as the earliest time step at which the algorithm registered a detection, inside the fault window duration τ_w. In this work a successful detection is registered if the algorithm gives detections that persist for at least $P\%$ of the fault window. The threshold percentage $P\%$ is a tuning parameter that prevents algorithms that generate random detections from getting a high early detection score. In this work, the threshold percentage was set to $P\% = 75\%$. The early detection score also considers detections that occur soon after the end of a fault. In order to facilitate this functionality, the fault window duration τ_w is extended for a short time τ_{FP} after the fault ends, where a small positive score is given for a FP that occurs during this time. The extension time is a design parameter which in this paper was set to $\tau_{FP} = 5$ hours after the end of a leakage.

The early detection score is calculated by first finding the delay of the first detection in the defined fault window, given by $x = t_d - t_f$. Then the following sigmoid function is used to assign a score to this detection: $\sigma(x) = 2/\left(1 + e^{((\alpha/\tau_w)x)}\right)$, where $x \in \{0, \cdots, \tau_w\}$. The parameter $\alpha > 0$ is defined such that $\sigma(\tau_w) \approx 0$, while $\sigma(0) = 1$ for any value of $\alpha < \infty$. Here a value of $\alpha = 6$ was chosen. A detection that occurs outside the fault window τ_w is not included in the score. The *raw* early detection score is then given by the sum of the early detection function score for all the faults $f_j, j \in \{1, \cdots, n_f\}$ that occur in each scenario $i \in \{1, \cdots, n_{sc}\}$, as follows:

$$S^*_{ED} = \sum_{i=1}^{n_{sc}} \sum_{j=1}^{n_f} \sigma(x_{ij}). \tag{5}$$

The S^*_{ED} then needs to be normalized by the score of the ideal algorithm on this metric, given by S^{ideal}_{ED}. In contrast, the rest of the metrics have ideal scores equal to one, thus the calculated scores are already normalized. The *normalized* early detection score is then given by:

$$S_{ED} = S^*_{ED} / \left(S^{ideal}_{ED}\right) \tag{6}$$

4 Evaluation of detection algorithms

To demonstrate the usage of the LeakDB dataset, four detectors were used which can be considered as a baseline for testing other detectors: An "Ideal" detector which assumes perfect knowledge and returns correct detections, a "Null" detector which does not detect any faults at any scenario, a "Random" detector which detects faults at random time steps, and a Minimum Night Flow (MNF) detector developed by the authors, using three different detection parameters. To test these detectors the benchmark dataset for 'Hanoi' network was used, by simulating $n_{sc} = 500$ leakage scenarios. The duration of each scenario is one year, with a time step of 30 minutes.

The MNF detector analyzes the minimum night flows during the night, to detect anomalies on the input flow of the network. A moving window w is defined in order to calculate the minimum MNF (mMNF) during those days; this is defined as $\mathrm{mMNF}(k, w) \triangleq \min\{\mathrm{MNF}(\tau)\}$ subject to $\tau \in \{k - w - 1, \cdots, k - 1\}$. Then, the MNF of the k-th day $\mathrm{MNF}(k)$ is computed and the difference $e(k) = \mathrm{MNF}(k) - \mathrm{mMNF}(k, w)$ is calculated. A fault is detected if the difference $e(k)$ is greater than a threshold as follows: $e(k) > \lambda\ \mathrm{mMNF}(k, w)$. Both the w parameter and $\lambda \in (0, 1)$ are design parameters which can be chosen with the help of the LeakDB dataset. When the threshold is exceeded,

Detectors	**Score** (%)			
	R_{TP}	R_{TN}	$F_{tp,fp}$	S_{ED}
Ideal	100.00	100.00	100.00	100.00
Null	0.00	100.00	0.00	0.00
Rand	50.19	49.79	32.11	0.21
MNF 10%	68.98	66.96	50.01	45.88
MNF 20%	39.90	99.58	56.49	32.35
MNF 30%	32.87	99.91	49.36	26.80

Table 1: Example detectors and their scores when applied on the LeakBD dataset of the "Hanoi" network. With respect to the F-measure, the best classifier is the MNF with $\lambda = 20\%$.

an error flag is raised, and the measurement is excluded from the calculation of the mMNF, i.e., only the last w measurements that were not flagged as faults are considered. This helps in avoiding learning the fault after the period w has passed. The MNF detector was scored when applied to the LeakDB scenarios using three different threshold gains: $\lambda = \{10\%, 20\%, 30\%\}$, and a moving window w corresponding to 10 days. The results for all the detectors are shown in Table 1.

The results in Table 1 indicate that the MNF algorithm performs better than a random algorithm. The MNF detector performs poorly on the early detection metric, as it can only detect an anomaly in the data during the night hours when minimum flow conditions occur. The comparison of the MNF algorithm using different thresholds, shows that there is a trade-off between early detection and classification accuracy. In this work we do not provide an overall score to the detection algorithms, so the "best" threshold for this algorithm can only be defined with respect to a specific metric. However, to choose the "best" leakage detection algorithm, appropriate weights must be given to each of the provided metrics, or use multi objective optimization methodologies.

5 Conclusions and future work

In this work we have presented the Leakage Diagnosis Benchmark (LeakDB), a realistic dataset of leakages in water distribution networks which can be used to evaluate different leakage diagnosis algorithms. Specific metrics for the evaluation of the algorithms have been proposed and a number of detection algorithms were evaluated.

This benchmark is currently being expanded, as more networks, detection algorithms and features are added. Future work on the benchmark can include additional features such as: 1) considering various anomalies that are due to sensor measurements such as spikes in data, missing measurements, sensor faults and noisy measurements, 2) consider other system faults, such as failure of a pump, 3) include real-world networks and data, in which case an important task would be for experts to manually inspect the data and label the faults.

During the creation of the benchmark, many challenges arose that should be addressed in future work. Firstly the benchmark space requirements may be excessive especially for large networks. This can be addressed using different data formats or omitting redundant data. Additionally, the various scenarios created need to be automatically validated in terms of satisfying realistic hydraulic conditions. Finally, the selection of an overall evaluation metric for the detection algorithms must be investigated.

6 Acknowledgments

This work is partially funded by the European Union Horizon 2020 programme under grant agreement no. 739551 (KIOS CoE), and by the Interreg V-A Greece-Cyprus 2014-2020 programme, co-financed by the European Regional Development Fund (ERDF) and National Funds of Greece and Cyprus, under the project SmartWater2020.

7 Data availability

The source code of the toolkit to generate the Leakage Diagnosis Benchmark, as well as links to the datasets, can be found at: `https://github.com/KIOS-Research/LeakDB`.

References

[1] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "A novel video dataset for change detection benchmarking," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4663–4679, 2014.

[2] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. d. R. Millán, and D. Roggen, "The opportunity challenge: A benchmark database for on-body sensor-based activity recognition," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 2033–2042, 2013.

[3] A. Lavin and S. Ahmad, "Evaluating real-time anomaly detection algorithms – the Numenta Anomaly Benchmark," in *Proceedings of 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2015, pp. 38–44.

[4] M. D. Jolly, A. D. Lothes, L. Sebastian Bryson, and L. Ormsbee, "Research database of water distribution system models," *Journal of Water Resources Planning and Management*, vol. 140, no. 4, pp. 410–416, 2014.

[5] A. F. Emmott, S. Das, T. Dietterich, A. Fern, and W.-K. Wong, "Systematic construction of anomaly detection benchmarks from real data," in *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description - ODD '13*. New York, New York, USA: ACM Press, 2013, pp. 16–21.

[6] D. G. Eliades and M. M. Polycarpou, "Leakage fault detection in district metered areas of water distribution systems," *Journal of Hydroinformatics*, vol. 14, no. 4, pp. 992–1005, 2012.

[7] K. A. Klise, M. Bynum, D. Moriarty, and R. Murray, "A software framework for assessing the resilience of drinking water systems to disasters with an example earthquake case study," *Environmental Modelling & Software*, vol. 95, pp. 420–431, 2017.

[8] H. Yamauchi and W.-y. Huang, "Alternative models for estimating the time series components of water consumption data," *Journal of the American Water Resources Association*, vol. 13, no. 3, pp. 599–610, 1977.

[9] G. Forman and M. Scholz, "Apples-to-apples in cross-validation studies," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 1, p. 49, 2010.