Peak Anomaly Detection from Environmental Sensor-Generated Watershed Time Series Data

Byung Suk Lee, John Clay Kaufmann, Donna M. Rizzo, and Ijaz Ul Haq

University of Vermont, Burlington VT 05405, USA
bslee@uvm.edu, jckaufma@uvm.edu, drizzo@uvm.edu,ihaq@uvm.edu

Abstract. Time series data generated by environmental sensors are typically "messy," with unexpected anomalies that must be corrected prior to extracting useful information. This paper addresses automatic detection of such anomalies and discusses two lines of study for achieving efficient and accurate detection using AI techniques with a focus on *peak* anomalies. One study uses the classic knowledge-engineering process and the other uses a deep-learning method to mimic how a trained watershed scientist detects anomalies. These two approaches were applied to time series data collected from a research watershed in Vermont, U.S.A., and their performances were assessed with respect to detection accuracy and computational efficiency. The two approaches had different anomaly detection accuracy depending on the peak type. The knowledge engineering approach was readily tunable to achieve competitive or better detection accuracy while computationally far more efficient than the deep learning approach. Results indicate the advantage of using the two approaches in combination, while a more general study involving other watersheds' time series data would be needed.

Keywords: Peak anomaly detection, sensor-generated data, time series data, knowledge engineering, deep learning.

1 Introduction

Background and motivation. Anomaly detection from sensor-generated time series data is an important problem in many real-world applications for manufacturing, monitoring, and management of resources. Often associated with the Internet-of-Things, there has been a large body of work conducted in this area (see the surveys by Cook et al. 2019 [1] and Sgueglia et al. 2022 [2]). Publications show a substantial focus on environmental sensor-generated time series data such as temperature, humidity, etc. (e.g., Hill and Minker 2010 [3], Jae-Myoung et al. 2020 [4], Conde 2011 [5], Hayes and Capretz 2014 [6], Hill and Minker 2006 [7], Russo et al. 2020 [8]).

This paper focuses on a data-driven approach to anomaly detection using watershed environment sensor-generated time series data. These time series data are typically "messy", with unexpected phenomena. At the present time, watershed scientists rely on manual examination of the time series data, their domain expertise in having observed a variety of anomaly types, and field notes during on-site sensor maintenance to detect anomalies. Recently there has been increasing attention in the environmental science community in replacing human effort with a more automated process (e.g., Jones et al., 2021 [9]). This is a challenging endeavor because of the "messiness" of data; but successful automation has the potential to bring great scientific and societal benefits.

Objectives. Our primary objective is to develop an automatic mechanism for detecting anomalies in time series data for a hydrological and biogeochemical study. A secondary objective is to build an inventory of common anomaly types for domain scientists (e.g., hydrologists) and study the performance for different anomaly types.

Methodology. The performance study employs classification-based anomaly detection, which is a supervised machine-learning task and requires labeled anomaly data for training and tuning. Two machine-learning approaches are used: knowledge-engineering and deep-learning. Knowledge engineering is a process that identifies parameters based on the domain expert to solve the problem at hand (i.e., detecting anomalies). The deep-learning fits a model, i.e., a neural network trained using expert labeled data. Knowledge engineering specializes in a fixed set of anomaly types (i.e., patterns) and uses a small number of "hand-crafted" parameters (e.g., 2 to 5 for each anomaly type) to characterize the patterns. Deep learning, on the other hand, generalizes to a variety of anomaly types (i.e., patterns) and uses millions of neural-network parameters (or coefficients) to characterize the patterns at different levels of the network. Naturally, knowledge engineering incurs much lower computing cost (in terms of both computation time and the memory consumption) to tune the parameters, and it may perform better or worse than deep learning depending on the complexity of the pattern.

Outcome summary. The deep learning approach and the knowledge engineering approach had different performances. They achieved different detection accuracies depending on the peak type, while overall accuracy was comparable. Besides, there were contrasting relative accuracies between the different time series data (fDOM and turbidity). The knowledge engineering was significantly more efficient computationally (i.e., training time and memory usage) for all peak types. Using both the knowledge engineering and the deep learning approaches in combination would take advantage of the different performances of the two approaches.

Contributions. The contributions of our work can be summarized as follows.

- It introduces peak anomalies as an important anomaly type and identifies a set of peak types of practical importance in hydrological watershed science.
- It implements classification-based peak anomaly detection using supervised learning techniques via knowledge engineering and deep learning approaches and compares the resulting performances.
- It identifies and labels the peak types in hydrological time series data collected from a watershed.

2 Related Work

There are different anomaly types handled by anomaly detection methods, categorized into *point* anomaly, *pattern* anomaly, and *system* anomaly (Lai et al., 2021 [10]), (Chandola et al., 2009 [11]). A point refers to a single time series sample data; a pattern is identified over a sequence of time series samples that exhibit a given characteristic (e.g., statistic, shape) or behavior (e.g., trend, change); and a system refers to a group of systems (e.g., multivariate time series patterns) where one of many systems is in an abnormal state.

Most of the existing work on anomaly detection addresses *point* anomalies (Cho et al., 2015 [12], (Enikeeva et al., 2019 [13]), (Fearnhead et al., 2010 [14]), (Fryzlewicz and Piotr, 2014 [15]), (Tveten et al., 2020 [16]). Pang et al., 2021 [17] mentioned that the methods for detecting point anomalies cannot be applied to group anomalies as they have entirely distinct characteristics. Group anomalies refer to a subset of anomalous data instances, which has the same definition as pattern anomalies used in our work. The peak anomalies in our watershed data are a type of pattern anomaly identified by the shapes of time series sample sequences.

There are some works on pattern anomaly detection from hydrological watershed sensor-generated time series data, mainly focused on detecting pattern deviations. Yu et al. ,2020 [18] used a distance-based approach to extract the trend and mean feature of time series segments of equal size, for which they proposed two algorithms called the Trend Feature Symbolic Aggregate approximation (TFSAX) and weighted Probabilistic Suffix Tree (wPST). Sun et al., 2017 [19]'s work also depends on significant feature points in which the time series is separated into numerous patterns; the system then calculates the pattern features using a density-based anomaly detection algorithm. Qin et al., 2019 [20] proposed the iForest algorithm to extract anomalous patterns using an adaptive segmentation algorithm based on key feature points; the pattern features of each time series segment are then translated to a k-dimensional space, i.e., restricted to a space with k orthogonal axes. The nearest neighbor distance is then used to extract top-K patterns and the K patterns with the highest anomaly scores are output. However, the pattern anomalies detected by these algorithms are not the peak anomalies handled in our work.

We find works closer to the peak anomaly detection in our time series data in other application domains, such as ECG anomaly detection (Lin et al., 2018 [21]), (Li et al., 2020 [22]). These ECG datasets are annotated with codes that indicate whether segments are normal or abnormal at each R peak location. However, to the best of our knowledge, there is nothing similar in hydrological watershed time series data.

3 Time Series Data and Peak Anomaly Types

3.1 Time series data

This study uses experimental sensor time series data collected over nine years at a small forested research watershed in Vermont, U.S.A.[1] The researchers measure stream stage, from which stream discharge is computed, at a 5-minute interval. They measure turbidity and fluorescent Dissolved Organic Matter (fDOM) at a 15-minute interval, using optical Turner Designs Cyclops sensors (see **Fig. 1**). The sensors are positioned below the depth of ice formation and are operated year-round. The data are used to estimate stream fluxes of dissolved and particulate organic carbon (DOC and POC). Turbidity in the water interferes with light transmission needed for the fDOM measurement, so fDOM values are corrected based on the turbidity values. Fluorescence is temperature sensitive, so fDOM values are also adjusted using concurrent water temperature measurements. The stage time series has 231,465 samples, and the turbidity and fDOM time series have 229,620 samples each.

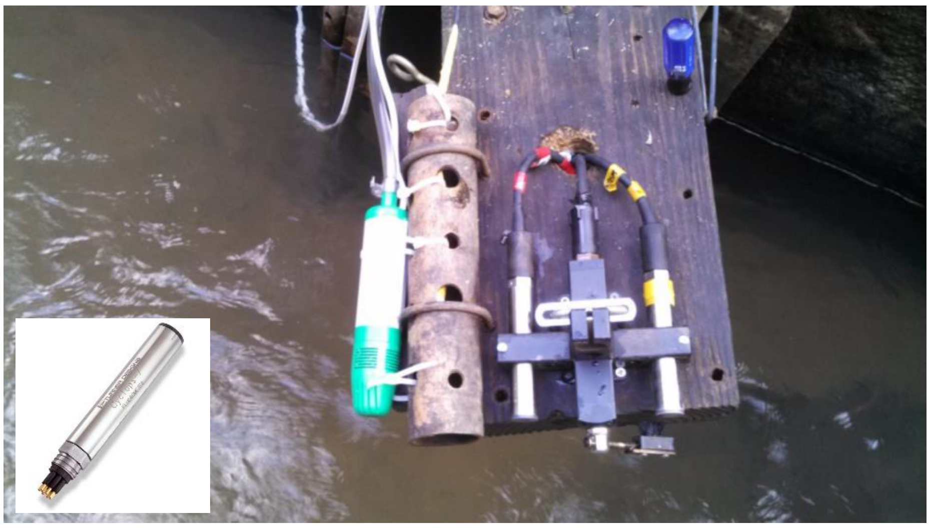
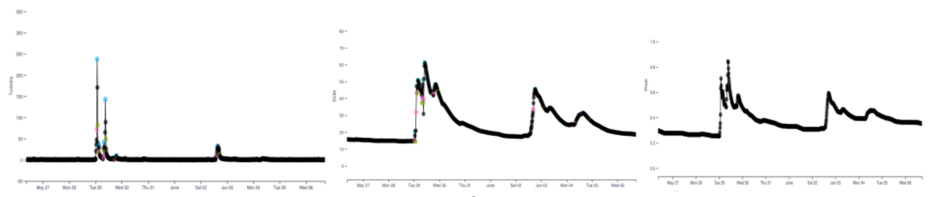


Fig. 1. Turbidity/fDOM sensor mounted on a board immersed in the water. The image in the corner is a Turner Designs Cyclops-7 submersible sensor.

Stage is already corrected and used as reference data; so, in this study, anomalies are defined and detected for fDOM and turbidity. The actual anomalies in the fDOM and turbidity data sets have been labeled through a visual examination and have been vetted by a domain scientist. We refer to these labeled datasets as the "ground truth" in this study. **Fig. 2** shows the three time-series (stage, fDOM, turbidity) segments with normal peaks.

[1] The watershed name is not mentioned due to a data management policy of the agency that owns the datasets.

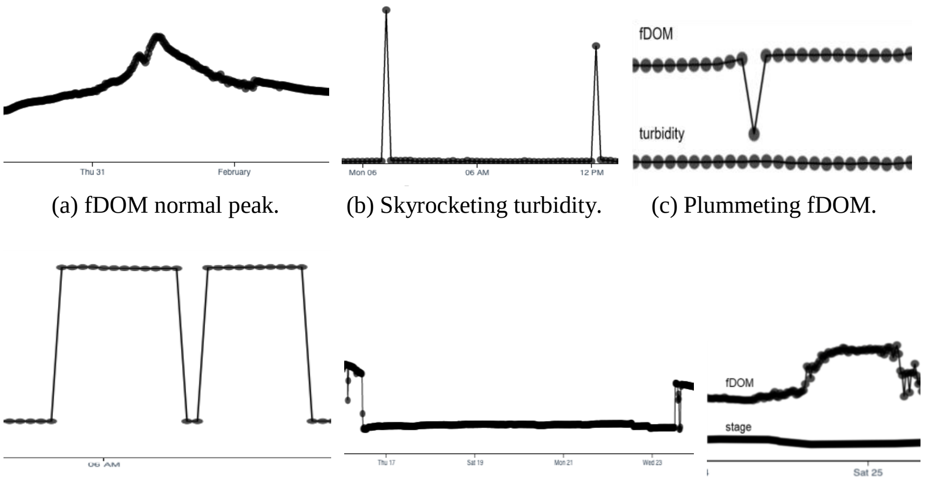


(a) Stage. (b) fDOM. (c) Turbidity.

Fig. 2. The watershed time series data (from May 27 – June 6, 2012) containing normal peaks.

3.2 Peak anomaly types

The focus is on "peak anomaly" types in this study. Five main anomaly types have been identified: skyrocketing peaks (SKP), plummeting peaks (PLP), flat plateau (FPT), flat sinks (FSK), and phantom peaks (PHP). Normal (i.e., non-anomalous) peaks (NAP) are of another peak type. See **Fig. 3** for illustrations. A skyrocketing peak is an upward spike or a narrow peak (with a short base width), and a plummeting peak is a downward spike; both may be caused by electronic sensor noise. A plummeting peak is observed in fDOM only and its detection requires that there is no preceding rise in the turbidity (which triggers a drop in fDOM). A flat plateau and a flat sink are characterized by near-constant signal amplitude near the top (plateau) and the bottom (sink); they may be caused by sediment deposition near or around the sensors. Flat sinks are observed in fDOM only. A phantom peak appears as a normal peak but has no preceding stage rise that triggers the peak; it may be caused by a non-hydrological event like animal activity in the water near the sensor. Note that detecting a phantom peak and a plummeting peak requires identifying causal relationships between two data time series, whereas the other peak types require only one data time series.



(a) fDOM normal peak. (b) Skyrocketing turbidity. (c) Plummeting fDOM.

(d) Flat plateau fDOM. (e) Flat sink fDOM. (f) Phantom fDOM.

Fig. 3. Examples of peak types.

4 Computational Methods

4.1 Knowledge engineering

As mentioned earlier, the knowledge engineering in this study is a process to emulate what a domain expert does to identify and detect anomalies. The knowledge gained from our expert hydrologist has been formalized into the definition of each anomalous peak type. The detection mechanism of anomaly instances according to the definition of an anomalous peak type and the associated threshold parameters are summarized below. The threshold parameters are tuned against the labeled anomaly instances in the ground truth time series. **Fig. 4** summarizes some key terms used in the definition of peak anomalies, where base width is the interval (number of data points) between the start time and end time of a peak; amplitude is the maximum rise above baseline among all values between the start and end times of the peak; and prominence is the amplitude of a peak measured relative to the amplitudes of the neighboring peaks.

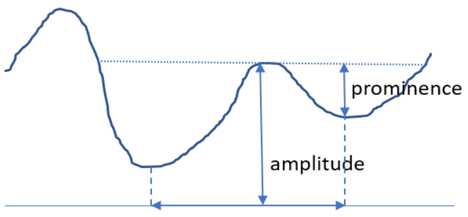


Fig. 4. Terms used in defining a peak.

Definition of peak anomaly types
The intuitive meaning, detection mechanism, and the associated threshold parameters are defined below for each anomalous peak type.

Skyrocketing peak (SKP) is an upward peak for which the base width is smaller than a threshold number δ_{SPBW} (e.g., 10) of data points and the prominence is larger than a threshold unit δ_{SPA} (e.g., 10 units). A probable cause is the sensor impulse noise.

Plummeting peak (PLP) (for fDOM only) is a downward peak (decrease) for which the base width is smaller than a threshold number δ_{PLPFBW} (e.g., 4) of points and the negative prominence (decrease from baseline) is larger than a threshold unit δ_{PLPFA} (e.g., 4 units), and there is no abrupt rise or elevated level of turbidity more than δ_{PLPTI} NTU (e.g. 100 NTU) within the preceding threshold time interval δ_{PLPFI} (e.g., 1 hour). Here, NTU stands for "Nephelometric Turbidity Units." A probable cause of a plummeting peak is the sensor impulse noise.

Flat plateau (FPT) denotes consecutive samples between an abrupt rise and an abrupt drop where the rise amplitude is more than a threshold value δ_{FPA} (e.g., 300) and the values within the plateau portion remain near-constant. Here, "near-constant" means (max − min) / max ratio of the sample amplitudes is less than a threshold ratio δ_{FPR} (e.g., 20%), and "abrupt" rise/drop is defined as a rise/drop of more than 10 units over the course of four points in the time series, before and after the ends of the plateau. A flat plateau may be caused by a sensor partially buried under sediment.

Flat sink (FSK) denotes consecutive samples between an abrupt drop and an abrupt rise where the drop amplitude is more than a threshold value δ_{FSA} (e.g., 2) and the amplitude within the drop portion is near-constant. Here, "near-constant" and "abrupt" are defined the same as those for the flat plateau (FPT). A flat sink may be caused by a sensor buried under sediment partially or completely.

Phantom peak (PHP) is an upward peak that is not preceded by a rise in stage within a threshold interval $\delta_{PHPI\text{-}F}$ for fDOM, $\delta_{PHPI\text{-}T}$ for turbidity (e.g., 0.5 hour) (and, hence, is not a real peak). There are additional conditions to account for exceptional cases specific to either fDOM or turbidity:

- For a phantom peak in fDOM, the period of Sep-15 to Oct-31 is not considered. This additional condition during the foliage season in Vermont accounts for fluorescent DOC released from fallen leaves in the stream channel, which may cause in increase in fDOM without a hydrological driver. (For phantom peaks in fDOM, the fDOM time series is smoothed prior to this detection in order to keep locally fluctuating small peaks from being detected as phantom peaks.)
- For a phantom peak in turbidity, the prominence of the peak is above a threshold δ_{PHPTA} and there is no turbidity interference evident in fDOM. True turbidity (values above 100 NTU) causes a drop in fDOM by more than a threshold ratio in an otherwise rising fDOM trajectory; and this phenomenon indicates that the turbidity peak is a real peak.

Fig. 5 illustrates fDOM phantom peaks in relation to the stage time series. **Fig. 6** illustrates turbidity phantom peaks in relation to the stage time series (subfigure a) and the fDOM time series (subfigure b).

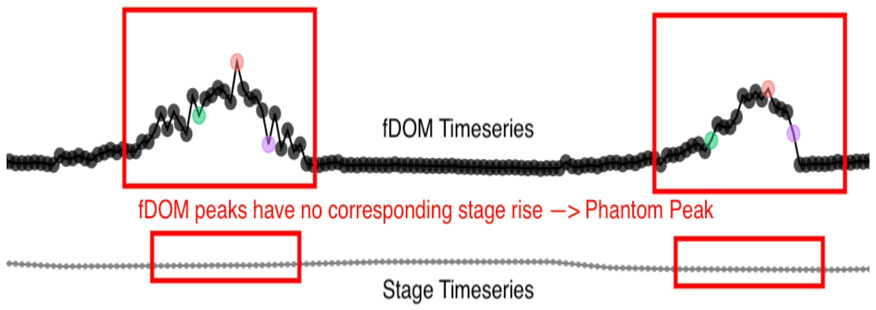
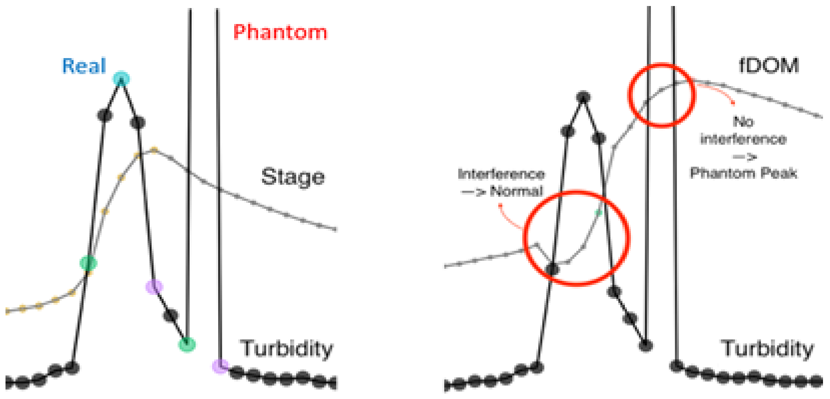


Fig. 5. fDOM phantom peaks.



(a) (b)

In the subfigure (a), the first peak following a stage rise is real, and the second peak is phantom. In the subfigure (b), the first peak causing a turbidity-interference in fDOM is real, and the second which does not is phantom.

Fig. 6. Turbidity phantom peaks.

Precedence among peak anomaly types. There are peaks that fit the definitions of two or more peak anomaly types. In this study, only one anomaly type is chosen according to a predefined precedence rule. Specifically, for fDOM, in order of highest to lowest precedence, skyrocketing peaks, phantom peaks, plummeting peaks, flat plateaus, flat sinks, non-anomaly peaks. For turbidity, the order is skyrocketing peaks, phantom peaks, flat plateaus, and non-anomaly peaks.

4.2 Deep learning

We have chosen ResNet as the deep learning model for its proven ability to avoid the vanishing gradient issue, thereby achieving outstanding classification accuracy. Specifically, we use a 1-D time-series ResNet-50 implementation, with the PyTorch model code obtained from the repository of Hong et al. 2020 [23]. It showed the best performance in a review by Fawaz et al. 2019 [24] for univariate time series classification among the current state-of-the-art deep learning-based time series classification algorithms.

Fig. 7 shows the network architecture of the deep learning model. The architecture has a $t \times 3$ input matrix, where t is the variable number of data samples, and the 3 represents the three types of time series data (i.e., fDOM, turbidity, and stage). As different peaks occur over varying sample lengths, we make use of PyTorch's pad_sequence function, which pads a list of variable length tensors with a given padding value. Specifically, we use a padding value of 0. We believe this is the least intrusive value possible, which is important as we do not want to give the classifier any extra bias from the padded values. The core idea behind the ResNet model is to use residual blocks that have shortcut connections between blocks to calculate the residual function, which eases learning as compared to much deeper convolutional neural networks (Hong et al. 2020 [23]).

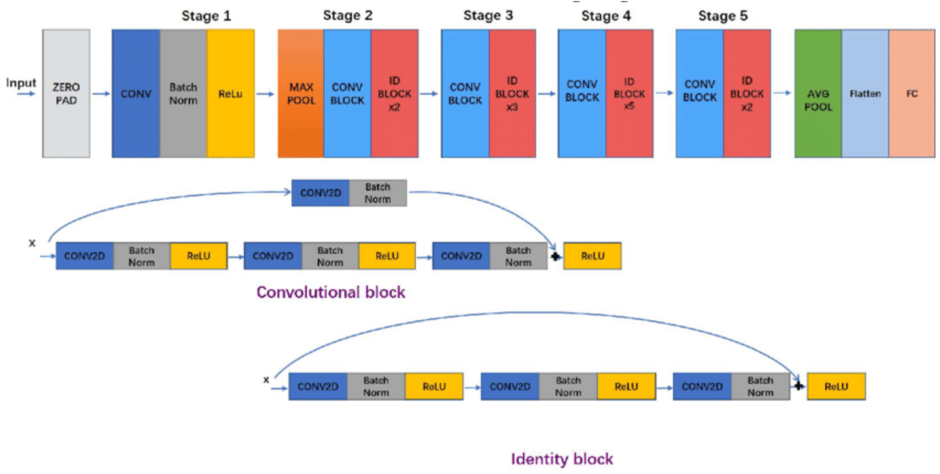


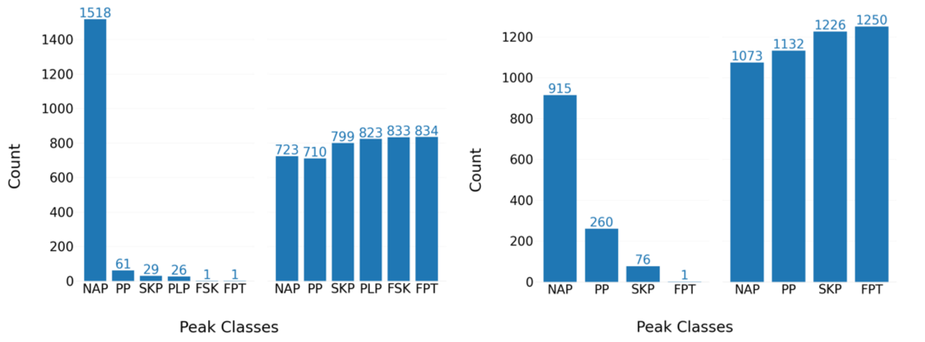
Fig. 7. Time series 1-D ResNet-50 architecture.

5 Performance Evaluation

5.1 Experiment setup

The anomaly detection framework used is multiclass classification. In the knowledge engineering approach, there is one binary classifier run for each anomaly type; the anomaly detection algorithm runs the multiple classifiers sequentially (not in parallel) to monitor individual time series, and, for each peak, looks at every classifier's response and determines the anomaly class. When the multiple classifiers detect different anomaly types, the precedence rule is applied to choose one. In the deep-learning approach, the three time-series -- stage, turbidity, and fDOM -- are treated as one tri-variate time-series, as mentioned in Section 4.2.

Data augmentation for class balancing. The dataset collected from the research watershed is severely skewed in the anomaly class distribution, with the non-anomalous peak (NAP) class accounting for 93% of the fDOM peaks and 73% of the turbidity peaks within the seven-year period of data between 2012 and 2019 (see **Fig. 8**(a) and (c)). Such a severe class imbalance would drive the classifier to focus on correctly detecting the far more numerous non-anomalous peaks rather than the far fewer anomalous peaks during training, as a result not trained adequately to detect anomalous peaks. So, we augmented the dataset to keep the class sizes better balanced (see **Fig. 8**(b) and (d)). The augmentation scheme alters randomly selected real peaks in both the peak base widths and the peak amplitude to a degree randomly selected within a predefined range. This range varies by the peak type. The peak amplitudes were multiplied by a uniformly generated number in the range of 1.1 to 3 for flat plateaus, 0.2 to 3 for flat sinks, and 0.8 to 1.2 for the other peak types. To produce a balanced peak distribution, the augmentation algorithm tracks the current distribution of peaks, and samples a new peak based on the current distribution. In addition, test-time augmentation was used for statistical significance of the result.



(a) fDOM original (b) fDOM augmented (c) Turb. Original (d) Turb. Augmented

Fig. 8. Class sizes of fDOM peaks and turbidity peaks before and after augmentation.

Training, validation, and testing scheme. We used prequential evaluation (as opposed to the conventional cross-validation) to consider the effect of temporal ordering inherent in time series. Basically, each new batch of data is first used as test data and then appended to the existing training data. Thus, the training data size keeps increasing, and so does the training time (see ***Fig. 9***). Our prequential evaluation is a "growing window" version adopted from the empirical study done by Cerqueira et al., 2020 [25]. Each time series data was split into 90% for training/validation and 10% for testing. The batch size was set to 32 samples.

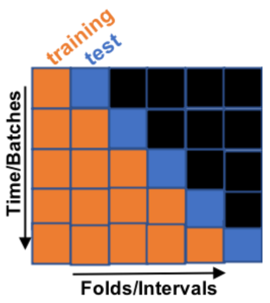


Fig. 9. Prequential evaluation (source: Cerqueira et al. [25]).

Knowledge engineering threshold parameter tuning. For each anomaly type, the random search approach was used to select parameter values in the parameter space defined by the threshold parameters belonging to the anomalous peak type. The set of parameter values that maximizes the anomaly detection performance was found using a random search iterated 1,000 times for each batch of data (added in the prequential evaluation). 1,000 iterations are more than enough, and it gives 99.996% probability of achieving near optimum within 1% of the true optimum. (A random search of n iterations has $1 - (1 - \varepsilon)^n$ probability of finding parameter values achieving near-optimum within the error ε from the true optimum (Firebug 2016 [26]).)

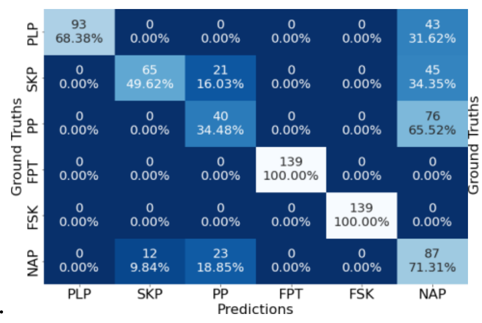
Deep learning model parameter tuning. Given the ResNet-50 used as the core model, the Adam optimizer was used, alongside a batch size of 32 samples, with 50 epochs and a learning rate of $1 \times e^{-3}$. Learning rate decay was added, with the learning rate decreasing by 0.1 every ten epochs. Early stopping was implemented as well, with a patience of five epochs. If the validation score did not change after five epochs, the algorithm stops training.

Performance metrics are the accuracy of peak anomaly detection and, additionally, computing resources (time and memory) consumed for the anomaly detection.

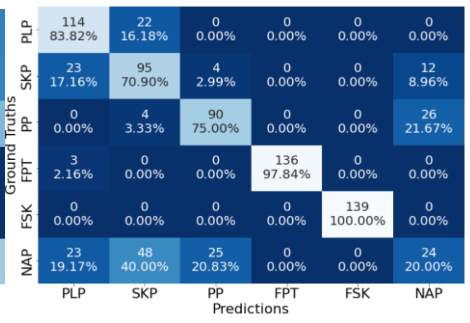
Computing platform. All experiments were performed on a local desktop, equipped with an i7 quad-core 4790k CPU clocked at 4.0 GHz, 16 GB of DDR3 RAM, a GeForce GTX 1080 GPU, and 750 GB of SSD storage.

5.2 Experiment results

Accuracy results. **Fig. 10** and **Fig. 11** show the confusion matrices of peak anomaly detection accuracy achieved by the knowledge engineering and the deep learning approaches for fDOM and turbidity time series, respectively.

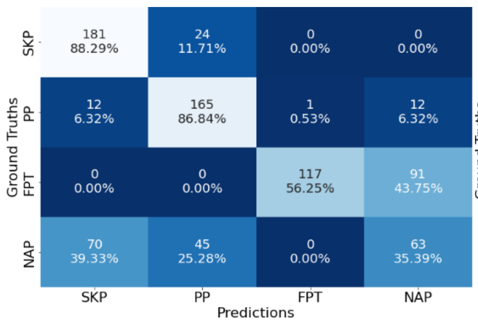


(a) Knowledge engineering

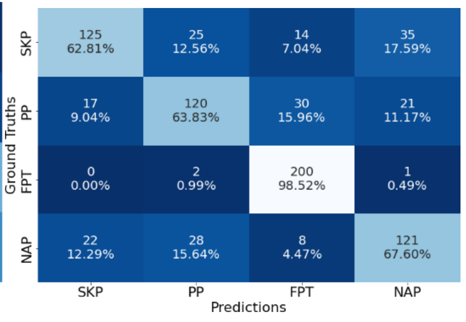


(b) Deep learning

Fig. 10. Confusion matrix of fDOM peak anomaly detection (PLP = plummeting; SKP = skyrocketing; PHP = phantom; FPT = flat plateau; FSK = flat sink; NAP = non-anomalous).



(a) Knowledge engineering



(b) Deep learning

Fig. 11. Confusion matrix turbidity peak anomaly detection (SKP = skyrocketing; PHP = phantom; FPT = flat plateau; NAP = non-anomalous).

Table 1 compares the accuracy achieved by the knowledge engineering approach and the deep learning approach (compiled from **Fig. 10** and **Fig. 11**) for each applicable peak type of fDOM and turbidity.[2]

Table 1. Comparison of accuracy between knowledge engineering and deep learning for fDOM (left) and turbidity (right).

fDOM	KE (%)	DL (%)	DL/KE ratio	KE/DL ratio
PLP	70.59	83.82	1.19	0.84
SKP	50.00	70.90	1.42	0.71
PHP	33.62	75.00	2.23	0.45
FPT	100.00	97.84	0.98	1.02
FSK	100.00	100.00	1.00	1.00
NAP	70.83	20.00	0.28	3.54

Turbidity	KE (%)	DL (%)	DL/KE ratio	KE/DL ratio
SKP	88.29	62.81	0.71	1.41
PHP	86.84	63.83	0.74	1.36
FPT	56.75	98.52	1.74	0.58
NAP	35.39	67.60	1.91	0.52

Additionally, **Table 2** shows the composite accuracy (i.e., balanced accuracy and F1-score) achieved for fDOM and turbidity.

Table 2. Composite accuracies achieved.

Approach	fDOM		Turbidity	
	Balanced accuracy (%)	F1-score (%)	Balanced accuracy (%)	F1-score (%)
Knowledge engineering	69.86%	67.74%	66.87%	66.01%
Deep learning	74.66%	72.05%	73.19%	71.23%

Computing costs. **Table 3** shows the computing costs of each approach in terms of the training time and memory usage. The computation time was clock time measured using a Python's built-in time package called "datetime". The clock was started upon initiation of the split 1 and was stopped upon completion of the split 5. Memory usage tracking was done using a Python's built-in memory tracker called "tracemalloc" for CPU memory in knowledge engineering and the PyTorch CUDA memory summary function for GPU memory in deep learning. The memory usage tracking started and stopped at the same clock points as the computation time tracking.

Table 3. Computing resources consumed. The average memory usage was calculated over the entire run time.

Approach	Training time	Average memory usage	Peak memory usage
Knowledge engineering	1.03 hours	1.39 MB	3.81 MB
Deep learning	4.08 hours	2001.34 MB	3015.57 MB

[2] Note that anomalous peaks are positive instances, and non-anomalous peaks are negative instances in this anomaly detection problem.

5.3 Discussion

The knowledge engineering approach and the deep learning approach had different anomaly detection accuracies depending on the peak type (**Table 1**), while the deep learning did a bit better than the knowledge engineering overall (see **Table 2**). The knowledge engineering approach was computationally a lot more efficient (consuming less computation time and memory) than the deep learning (see **Table 3**). Let us share below some further observations made.

Accuracy. Summarizing the confusion matrices (**Fig. 10** and **Fig. 11**) gives interesting contrasts in the peak anomaly detection accuracy between fDOM and turbidity (see **Table 1**).

- For fDOM, deep learning outperformed knowledge engineering for PLP (by 1.23 times), SKP (1.43 times), and PHP (2.18 times) while comparable for FPT and FSK. On the other hand, knowledge engineering outperformed deep learning for NAP (i.e., normal peaks) by 3.57 times.
- For turbidity, the results are different and somewhat reversed. Knowledge engineering outperformed deep learning for SKP (by 1.41 times) and for PHP (1.36 times) whereas underperformed for FPT (1.74 times) and NAP (1.91 times).

It is particularly noticeable that the deep learning did worse with NAP for fDOM while better with NAP for turbidity. We speculate this difference is due to their being more anomaly types for fDOM (five) than for turbidity (three), and therefore when being trained for fDOM the deep learning "paid more attention" to correctly detecting anomalous peaks at the expense of detecting non-anomalous peaks incorrectly.

Additionally, for turbidity, there was a large gap in accuracy for FPT between deep learning and knowledge engineering. We speculate the underlying cause is an excessive sparsity of FPT instances in the dataset. The original turbidity dataset has only one labeled FPT instance, and this single instance does not seem as steep (along the rising and falling edges of the plateau) as the knowledge engineering approach was looking for. Although during the peak augmentation (see Section 5.1) some of the FPT instances may have been made steep enough, not all of them would have been; and, as a result, the limited number of parameters for knowledge engineering caused the approach to miss the overall shape of the plateau, something that ResNet was extremely successful at (note that the accuracies for FPT and FSK in *fDOM* were quite high).

Computing resources. As shown in **Table 3**, the knowledge engineering approach used much less computation time (25%) and memory (0.36% on average, 0.13% peak) for the threshold parameter tuning than the deep learning approach for the model training (i.e., model parameter tuning). This is expected given the large difference in the number of parameters needed in the two approaches --- that is, a few threshold parameters per anomaly type in the knowledge engineering as opposed to over 25 million parameters (see Table 8 in Zagorukyo and Komodakis, 2016 [27]) in the deep learning. Note that the amount of computing resources used by the knowledge engineering is within the control of the “knowledge engineers” doing the parameter tuning and, specifically, depends on the number of random search points tried during the tuning; the results in **Table 3** are for 1,000 random search points.

Combining knowledge engineering and deep learning. Given the differences in accuracies observed for the knowledge engineering and the deep learning for different peak types, combining the two approaches toward improving the accuracy would be a natural next step. One straightforward way is to use both approaches, and for each peak type, choose the approach that had the higher accuracy in the test results. For example, with fDOM, we would choose the deep learning approach for all peak types except for NAP, for which we would choose the knowledge engineering approach. Ultimately, integrating the two approaches into a single classifier would achieve the best accuracy, and we defer this to the future work.

6 Conclusion

This paper presented a study conducted to detect peak anomalies from hydrological time series data collected from a watershed in Vermont, U.S.A. We identified a set of peak anomaly types important to the hydrological and geochemical study of the watershed and implemented two different computational approaches, knowledge engineering and deep learning. We then assessed performance between the two approaches with respect to the anomaly detection accuracy and the computational resources (time and memory). The differences between knowledge engineering and deep learning for fDOM and turbidity were quite interesting. For fDOM, we believe that the lower detection accuracy on non-anomalous peaks can be attributed to dealing with a larger number (five) of anomalous peak type classes, as compared with three in turbidity. This difference caused the fDOM deep learning classifier to be worse at detecting non-anomalous peaks.

There are a number of future works in the plan. First, we will further improve the computational tuning/training by breaking the time series data by season and model each season separately. The breakdown can be by the calendar, such as winter (December – March), spring (April – May), summer (June – September), and fall (October — November), or can be adaptive to the actual data by incorporating a time series change point detection algorithm (e.g., BEAST (Bayesian Estimator for Abrupt Seasonal and Trend change) (Kaiguang 2022 [28])). Second, we will generalize the study to include multiple additional watersheds, and extend the work toward automated machine learning which selects the best model (tuned parameters) based on precompiled characteristics of the input time series data (Chatterjee et al. [29]). Having more data from different geographical areas would also lead to a more generalized classifier, as different watersheds may exhibit peak instances of different shapes for the same anomaly type. Third, we will develop a mechanism involving some form of merged classifier that trains using both the deep learning approach and the knowledge engineering approach. A model could learn to rely upon one or the other for a given peak type, using confidence values or some other form of measurement, and train using this method. This could lead to higher accuracies in detecting anomalies in different peak classes.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. EAR 2012123. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government. The work was also supported by the University of Vermont College of Engineering and Mathematical Sciences through the REU program.

The authors would like to thank the US Geological Survey (USGS) for offering the domain expertise that was crucial to identify the peak anomaly types that are of practical importance.

References

1. Cook, A., Mısırl, G., Fan, Z.: Anomaly Detection for IoT Time-Series Data: A Survey. IEEE Internet of Things 7(7), 6481-6494 (2020).
2. Sgueglia, A, Di Sorbo, A., Visaggio, C., Canfora, G.: A systematic literature review of IoT time series anomaly detection solutions. Future Generation Computer Systems 134, 170-186 (2022)
3. Hill, D. J., Minsker, B. S.: Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. Environmental Modelling & Software 25(9), 1014(2010).
4. Kim, J.-M., Cho, Y.W., Kim, D.-H.: Anomaly detection of environmental sensor data using recurrent neural network at the edge device. In: Proceedings of the 2020 International Conference on Information and Communication Technology Convergence, 1624-1628. IEEE (2020).
5. Conde, E.: Environmental Sensor Anomaly Detection Using Learning Machines. MS Thesis. Utah State University (2011).
6. Hayes, M., Capretz, M.: Contextual Anomaly Detection in Big Sensor Data. In: Proceedings of the 2014 IEEE International Congress on Big Data, 64-71. Anchorage, AK, USA (2014).
7. Hill, D.J. and Minsker, B.S.: Automated fault detection for in-situ environmental sensors. In: Proceedings of the 7th International Conference on Hydroinformatics. Nice, France (2006)
8. Russo, S., Lürig, M., Hao, W., Matthews, B., Villez, K.: Active learning for anomaly detection in environmental data. Environmental Modelling & Software (134), 104869 (2020).
9. Jones, A.S., Jones, T.L., Horsburgh, J.S.: Toward automating post processing of aquatic sensor data. Environmental Modelling & Software (151), 105364 (2022).
10. Lai, K., Zha, D., Wang, G., Xu, J., Zhao, Y., Kumar, D., Chen, Y., Zumkhawaka, P., Wan, M., Martinez, D., Hu, X.: Tods: An automated time series outlier detection system. In:Proceedings of the aaai conference on artificial intelligence, Vol. 35. No. 18, pp. 16060-16062 (2021)
11. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), 1-58. (2009)

12. Cho, H., Fryzlewicz, P.: Multiple-Change-Point Detection for High Dimensional Time Series Via Sparsified Binary Segmentation. Journal of the Royal Statistical Society, 475–507 (2015)
13. Enikeeva, F., Harchaoui, Z.: High-Dimensional Change-Point Detection Under Sparse Alternatives. The Annals of Statistics, 2051–2079 (2019).
14. Fearnhead, P., Rigaill, G.: Changepoint Detection in the Presence of Outliers. Journal of the American Statistical Association, 169–183 (2019).
15. Fryzlewicz, P.: Wild Binary Segmentation for Multiple Change-Point Detection. The Annals of Statistics, 2243–2281 (2014).
16. Tveten, M., Eckley, I.A., Fearnhead, P.: Scalable change-point and anomaly detection in cross-correlated data with an application to condition monitoring. The Annals of Applied Statistics, 721-743 (2022).
17. Pang, G., Shen, C., Cao, L. Hengel, A.V.D.: Deep learning for anomaly detection: A review. ACM Computing Surveys (CSUR), 1-38 (2021).
18. Yu, Y., Wan, D., Zhao, Q., Liu, H.: Detecting pattern anomalies in hydrological time series with weighted probabilistic suffix trees. Water, 1464 (2020).
19. Sun, J., Lou, Y., Ye, F.: Research on anomaly pattern detection in hydrological time series. In:14th Web Information Systems and Applications Conference (WISA), pp. 38-43 IEEE (2017)
20. Qin, Y., Lou, Y.: Hydrological time series anomaly pattern detection based on isolation forest. IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 1706-1710. IEEE (2019).
21. Lin, Y., Lee, B., Lustgarten, D.: Continuous detection of abnormal heartbeats from ECG using online outlier detection. Annual International Symposium on Information Management and Big Data. Springer, Cham (2018).
22. Li, H., Boulanger, P.: A survey of heart anomaly detection using ambulatory Electrocardiogram (ECG). *Sensors* 1461 (2020).
23. Hong, S., Xu, Y., Khare, A., Priambada, S., Maher, K., Aljiffry, A., Sun, J., Tumanov, A.: Holmes: Health online model ensemble serving for deep learning models in intensive care units. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1614-1624. ACM, New York, USA (2020).
24. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. Data Mining and Knowledge Discovery 33, 917–963 (2019).
25. Cerqueira, V., Torgo, L, Mozeti ̌c, I.: Evaluating time series forecasting models: An empirical study on performance estimation methods. Machine Learning 109, 1997–2028 (2020).
26. Firebug: Practical hyperparameter optimization: Random vs. grid search. https://stats.stackexchange.com/q/209409 (2016)
27. Zagorukyo, S., Komodakis, N.: Wide Residual Networks. Proceedings of the British Machine Vision Conference, pp. 87.1-87.12. BMVA Press (2016).
28. Kaiguang: Bayesian Changepoint Detection & Time Series Decomposition. MathWorks, https://www.mathworks.com/matlabcentral/fileexchange/72515-bayesian-changepoint-detection-time-series-decomposition (2022)
29. Chatterjee, S., Bopardikar, R., Guerard, M., Thakore, U., Jiang, X.: MOSPAT: AutoML based Model Selection and Parameter Tuning for Time Series Anomaly Detection. arXiv preprint arXiv:2205.11755 (2022).