Machine Learning with Applications 16 (2024) 100543

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

An automated machine learning approach for detecting anomalous peak patterns in time series data from a research watershed in the northeastern United States critical zone☆

Ijaz Ul Haq [a,*], Byung Suk Lee [a,d], Donna M. Rizzo [b,d], Julia N. Perdrial [c,d]

[a] Department of Computer Science, University of Vermont, Burlington, VT 05405, USA
[b] Department of Civil and Environmental Engineering, University of Vermont, Burlington, VT 05405, USA
[c] Department of Geography and Geosciences, University of Vermont, Burlington, VT 05405, USA
[d] GUND Institute of the Environment, Burlington, University of Vermont, Burlington, VT 05405, USA

ARTICLE INFO

Keywords:
Automated machine learning
Anomaly detection
Deep learning models
Synthetic data
Time series data

ABSTRACT

This paper presents an automated machine learning framework designed to assist hydrologists in detecting anomalies in time series data generated by sensors in a research watershed in the northeastern United States critical zone. The framework specifically focuses on identifying *peak-pattern* anomalies, which may arise from sensor malfunctions or natural phenomena. However, the use of classification methods for anomaly detection poses challenges, such as the requirement for labeled data as ground truth and the selection of the most suitable deep learning model for the given task and dataset. To address these challenges, our framework generates labeled datasets by injecting synthetic peak patterns into synthetically generated time series data and incorporates an automated hyperparameter optimization mechanism. This mechanism generates an optimized model instance with the best architectural and training parameters from a pool of five selected models, namely Temporal Convolutional Network (TCN), InceptionTime, MiniRocket, Residual Networks (ResNet), and Long Short-Term Memory (LSTM). The selection is based on the user's preferences regarding anomaly detection accuracy and computational cost. The framework employs Time-series Generative Adversarial Networks (TimeGAN) as the synthetic dataset generator. The generated model instances are evaluated using a combination of accuracy and computational cost metrics, including training time and memory, during the anomaly detection process. Performance evaluation of the framework was conducted using a dataset from a watershed, demonstrating consistent selection of the most fitting model instance that satisfies the user's preferences.

1. Introduction

In-stream environmental sensors are now commonly deployed in various watersheds across the United States to monitor water quality. However, a common limitation in these studies is the delay between data acquisition and analysis, mostly due to the inability of many domain scientists to rapidly identify anomalies and clean large datasets efficiently. In this study, conducted as part of the NSF-funded Critical Zone Collaborative Network (CZCN) project, we present a case study of ecosystem data collected from sensors deployed at a watershed in Vermont, which serves as a testbed for our research. These sensors measure a variety of in-stream parameters, such as fluorescent dissolved organic matter (fDOM), turbidity, water level (to compute streamflow), and water temperature. The raw data from these sensors are messy and contain various anomalies. One particularly problematic type of anomaly in the project study is *peak-pattern* anomaly observable in a sequence of consecutive point measurements (i.e., time series samples), caused by a range of hydrological and non-hydrological events. After a year of review, domain scientists have identified and named these patterns. However, to analyze the data efficiently, cleaning is necessary either by removing or correcting those anomalies that are detected.

Anomaly detection in watershed time series data (WTSD) is crucial for effectively monitoring and managing water systems and resources. Anomaly detection in this context refers to identifying deviations from the standard, normal, or expected behavior in WTSD. These anomalies can provide valuable information about important events or may mislead the decision process. Detecting anomalies in WTSD is challenging

☆ This document is the results of the research project funded by the National Science Foundation, United States.
* Corresponding author.
E-mail addresses: ihaq@uvm.edu (I.U. Haq), bslee@uvm.edu (B.S. Lee), drizzo@uvm.edu (D.M. Rizzo), jperdria@uvm.edu (J.N. Perdrial).

https://doi.org/10.1016/j.mlwa.2024.100543
Received 8 December 2023; Received in revised form 28 February 2024; Accepted 5 March 2024
Available online 7 March 2024

due to the unpredictable nature of natural systems. Current methods typically focus on identifying single anomalous data points, known as point anomalies, without considering anomalies that span multiple points, known as pattern anomalies. These latter anomalies require the assessment of previous data points in relation to current data points, making their detection more complex. Therefore, there is a need for a reliable peak-pattern anomaly detection framework that can specifically detect and remove these repeating anomalous patterns.

Several use cases in the field of hydrology require accurate and efficient detection of pattern anomalies. For example, detecting and repairing anomalous peaks in dissolved organic carbon (DOC) data is necessary for accurate analysis of the concentration–discharge (C–Q) relation for DOC (Evans & Davies, 1998; Hamshaw, Denu, Holthuijzen, Wshah, & Rizzo, 2019; Vaughan et al., 2017). Additionally, detecting unusual patterns in streamflow data, such as flat lines or unmatched peaks, can aid in model calibration and better flood forecasting. Pattern anomaly detection in WTSD is also helpful in identifying sensor malfunctions and understanding the impact of seasonal and precipitation variations on hysteresis in C–Q relations.

Current trends for automating anomaly detection in WTSD use machine learning (ML) methods. However, determining the appropriate ML model can be challenging due to a large number of potential models available and the varying data characteristics of different watersheds. In order to address these issues, we propose the development of an end-to-end automated machine learning (autoML) pipeline called *Hands-Free Peak Pattern Anomaly Detection (HF-PPAD)*. HF-PPAD aims to provide an automated and effective solution for detecting pattern anomalies in WTSD, making it accessible and convenient for domain scientists. It requires a thorough understanding of anomaly detection algorithms for users to choose the right one, which often necessitates a strong background in generative models and statistical assumptions. Properly setting the parameters for these algorithms often requires a detailed understanding of their inner workings. Most domain scientists (often hydrologists and biogeochemists in this case) may not have such background, and HF-PPAD is designed to assist. HF-PPAD utilizes *supervised* deep learning models to deliver enhanced anomaly detection performance compared with other unsupervised or semi-supervised methods. In this work, we chose InceptionTime, MiniRocket, ResNet, TCN, and LSTM as our supervised deep learning models due to their strong results in various machine-learning tasks (Ismail Fawaz et al., 2019). MiniRocket is a recently developed model that can extract features from time series data with high efficiency, making it suitable for large-scale datasets (Dempster, Schmidt, & Webb, 2021). ResNet is a widely recognized model known for its high accuracy and has been adapted for time series data analysis (Jing et al., 2021). InceptionTime (Ismail Fawaz et al., 2020), on the other hand, is specifically designed for analyzing time series data (Ismail Fawaz et al., 2019), and TCN has been shown to perform well in time series classification tasks and is lightweight, making it ideal for resource-constrained environments (Pelletier, Webb, & Petitjean, 2019). Additionally, our choice of LSTM was based on its proven effectiveness in a wide range of time series applications (Hochreiter & Schmidhuber, 1997). These models can be configured in a variety of ways, with ResNet, InceptionTime, and LSTM being highly capable, while MiniRocket and TCN are more lightweight options.

The HF-PPAD performs several tasks, including the generation of a synthetic labeled peak pattern anomaly dataset for WTSD, automating the generation of an optimal instance of each model in the given pool through hyperparameter optimization, and choosing the best model instance based on the user's relative preference between high accuracy and lightweight model. HF-PPAD employs a state-of-the-art time series data synthesis tool like TimeGAN (Yoon, Jarrett, & van der Schaar, 2019) to automatically generate a significant amount of time series data containing labeled peak pattern anomalies similar to the original peak-pattern anomalies; this greatly reduces the expensive overhead of labeling anomalous pattern instances in the original data for supervised learning. The model instance building and selection process utilizes hyperparameter optimization techniques such as random forest, HyperBand, Bayesian optimizer, and a greedy search technique (Feurer & Hutter, 2019; Senagi, 2019).

To the best of our knowledge, this work is the first to provide an automated peak pattern anomaly framework that performs comprehensive tasks ranging from the generation of a fully labeled peak pattern anomaly dataset needed for supervised training of anomaly detection in the absence of a ground truth labeled dataset. The method also automates the selection of the best model instance based on user's preference on the anomaly detection accuracy and the computational cost for the watershed time series dataset. In summary, the main contributions of this work are as follows.

1. To propose an end-to-end automated *peak* anomalous pattern detection framework, furthermore for watershed time series data.
2. To use TimeGAN to generate labeled synthetic watershed time series data and peak pattern anomalies.
3. To automatically generate (i.e., design and select) the best model instance (i.e., deep learning classifier) from a pool of models according to the user's preference between accuracy and model instance size.

The remainder of this paper is organized as follows. Section 2 reviews the literature related to our study. Section 3 categorizes the various peak-pattern anomalies observable in watershed data, while Section 4 delves into the practical applications of HF-PPAD. The methodology behind HF-PPAD, including data preparation and model selection processes, is detailed in Section 5. Experimental results are presented in Section 6, followed by an analysis of the benefits derived from employing a synthetic dataset in Section 7. The paper concludes with Section 8, summarizing our findings and suggesting directions for future research.

2. Related work

2.1. Peak anomaly detection

Anomaly detection in time series data is a multifaceted field, encompassing various types of anomalies such as point anomalies, pattern anomalies, and system anomalies (Chandola, Banerjee, & Kumar, 2009; Lai et al., 2020). While point anomalies represent irregularities in single data points, pattern anomalies, the focus of our study, are identified by sequences exhibiting atypical characteristics or behaviors (e.g., trends or changes). System anomalies involve abnormalities in a group of sequences or systems. Despite growing interest, much of the existing research in anomaly detection has predominantly focused on point anomalies, employing methods like statistical thresholding and clustering (Cho & Fryzlewicz, 2015; Enikeeva & Harchaoui, 2019). However, these techniques are not directly applicable to pattern anomalies, which require analysis of sequential data patterns. For instance, Fearnhead and Rigaill (2019) explored change-point detection in financial time series, but their approach does not adequately address the complexities of pattern anomalies in environmental data. Similarly, Tveten, Eckley, and Fearnhead (2022) introduced an algorithm for detecting multiple change-points in large datasets, but their method lacks the sensitivity needed for nuanced patterns in hydrological data.

In contrast, pattern anomalies like peak-pattern anomalies in hydrological time series data present unique challenges. These anomalies are identified by the shape and sequence of data points, often requiring more sophisticated analysis methods (Lee, Kaufmann, Rizzo, & Haq, 2023). Efforts to detect pattern anomalies in hydrological data (Qin & Lou, 2019; Sun, Lou, & Ye, 2017; Yu, Wan, Zhao, & Liu, 2020) have primarily focused on deviations from established patterns, yet do not adequately address peak anomalies. For example, Sun et al. (2017) developed a method for detecting irregular patterns in river flow

data, but their technique does not differentiate between types of peak anomalies, which is crucial for our research.

Interestingly, more relevant work on peak anomaly detection is found in other domains, such as ECG anomaly detection. The work by Lin, Lee, and Lustgarten (2018) and Li and Boulanger (2020) on ECG datasets provides insights into handling time series data with annotated peak anomalies. These studies offer valuable methodologies for identifying and classifying different types of peak anomalies based on shape and sequence, which could be adapted for our purposes.

While the field of anomaly detection in time series data is well-established, there is a noticeable gap in research specifically addressing peak-pattern anomalies in hydrological data. Our work aims to bridge this gap by adapting and extending methodologies from other domains, such as ECG data analysis, to the context of hydrological time series, providing a more nuanced and effective approach to peak anomaly detection (Kulanuwat et al., 2021).

2.2. Automated machine learning in hydrology

Automated Machine Learning (AutoML) represents a significant paradigm shift in the application of machine learning (ML) in hydrology, a field that has leveraged ML techniques for over seventy years (Dramsch, 2020). The primary challenge in hydrological ML applications has been the selection of appropriate models for specific datasets and problems, a task traditionally demanding significant domain expertise (Ghobadi & Kang, 2023; Schmidt, Heße, Attinger, & Kumar, 2020).

AutoML emerges as a solution to this challenge by automating the process of model selection and optimization (Wu, Xi, & He, 2022; Yao et al., 2018). It simplifies the model-building process and democratizes ML use, making it accessible even to non-experts. However, despite these advantages, the application of AutoML in hydrology is still in its infancy. Current methods focus on optimizing models for narrowly defined problems or datasets, often overlooking the broader applicability required in hydrology (Ho & Goethals, 2022; Khan, Khan, & Alharbi, 2020).

Current AutoML tools like Auto-WEKA (Kotthoff, Thornton, Hoos, Hutter, & Leyton-Brown, 2019) and Auto-Sklearn (Feurer et al., 2015) have made significant advancements in automating model selection and hyperparameter tuning. Yet, they primarily serve traditional ML approaches and lack support for deep learning models, which are crucial for complex hydrological datasets. The emergence of AutoKeras (Jin, Song, & Hu, 2019) represents significant progress in automating the optimization of deep neural networks, primarily in the realms of text and image data. However, its application to hydrological time series analysis may require specialized data preprocessing and model configuration to accommodate the unique characteristics of time-series data.

Our work addresses these gaps by developing an AutoML pipeline specifically tailored for deep learning in hydrological time series analysis. This pipeline goes beyond the traditional scope of AutoML tools by incorporating a range of deep learning architectures and training hyperparameters, along with advanced optimization strategies like random forest, Bayesian, Hyperband, and greedy search algorithms. In doing so, it enhances model selection accuracy and optimizes computational efficiency, crucial for processing large hydrological datasets (Prasad et al., 2022; Sit et al., 2020).

Furthermore, our framework represents a application of AutoML in transforming peak-pattern anomaly detection in WTSDs from a complex, expert-driven task to a more accessible, automated process. This innovative approach converts anomaly detection to a supervised multi-class classification task, enabling more accurate and efficient identification of anomalies in hydrological data (Shen, Chen, & Laloy, 2021).

2.3. Unsupervised/semi-supervised versus supervised anomaly detection

In the diverse landscape of anomaly detection, the distinction between unsupervised, semi-supervised, and supervised learning methods is pivotal, especially in the context of time series data (Deng & Yu, 2014; Khan, Niu, Nyamawe, & Haq, 2021). Unsupervised and semi-supervised methods have gained significant traction, largely due to the difficulty in obtaining labeled datasets for anomalies. These approaches, including clustering, LSTM-based regression, autoencoders, and GANs, excel in identifying unknown patterns without pre-labeled data (Bahri et al., 2022; Ergen, Mirza, & Kozat, 2017; Schmidl, Wenig, & Papenbrock, 2022).

However, their ability to distinguish between normal variability and true anomalies can be limited, often leading to reduced precision. Unsupervised learning, in particular, demands substantial computational resources, which can impede its practicality for large-scale applications (Bahri et al., 2022; Zhu, Wu, & Liu, 2023).

The introduction of AutoML into unsupervised learning tasks has significantly automated the detection of anomalies, including point anomalies and change-points, across various domains. Systems such as PyOD (Zhao, Nasrullah, & Li, 2019), PyODDS (Li, Zha, Venugopal, Zou, & Hu, 2020), MetaAAD (Zha, Lai, Wan, & Hu, 2020), and TODS (Lai et al., 2021) represent substantial advancements in this field, facilitating broad-spectrum anomaly detection capabilities. Despite these advancements, the emphasis of these systems on unsupervised learning scenarios poses limitations, particularly in the domain of hydrological time series analysis, as detailed in our research (Lee et al., 2023). Peak-pattern anomalies inherent to hydrological data, characterized by their requirement for nuanced interpretation and contextual understanding, challenge the generalized models employed by these existing AutoML systems. This limitation is evident in the comparative analysis presented in Table 1, which highlights the unique capabilities of our HF-PPAD framework in addressing these specialized anomalies through a tailored approach, leveraging supervised learning methods for enhanced precision and specificity. The specific nature of peak-pattern anomalies in hydrological time series underscores the necessity for approaches that go beyond the capacities of current unsupervised learning frameworks, advocating for a system like HF-PPAD that is adept at handling such specialized tasks.

Pivoting towards supervised learning for peak-pattern anomaly detection, our research tackles the challenge of creating a labeled dataset for hydrological data. While the acquisition of such data can be demanding, it offers significant advantages in anomaly detection accuracy and specificity (Li, Jamieson, DeSalvo, Rostamizadeh, & Talwalkar, 2017; Ryzhikov, Borisyak, Ustyuzhanin, & Derkach, 2021). Constructing a domain-specific labeled dataset enables our model to achieve heightened accuracy and efficiency in detecting peak-pattern anomalies, addressing the shortcomings of current unsupervised and semi-supervised methods (Li & Boulanger, 2020).

Our work reveals the potential and advantages of supervised learning in anomaly detection, particularly for specialized tasks like peak-pattern anomaly detection in hydrological time series data. By moving beyond the limitations of existing AutoML systems, we demonstrate a more precise and efficient approach, underlining the emerging importance of supervised learning in this field.

3. Watershed data and peak-pattern anomaly types

3.1. Watershed time series data

Sensor data were collected from the study watershed over a period of eight years (from October 1, 2012 to October 1, 2020), encompassing a comprehensive dataset crucial for our analysis. Measurements of stream stage, turbidity, and fluorescent dissolved organic matter (fDOM) were taken at regular intervals—every 5 min for stream stage and every 15 min for turbidity and fDOM. These measurements were

Table 1
Comparison of anomaly detection frameworks with a focus on hydrological time series data analysis.

Feature	HF-PPAD (our Work)	TODS	PyOD	PyODDS	MetaAAD
Domain Focus	Hydrological time series data	General time series data	Multivariate data	Online data streams	Meta-learning for anomaly detection
Anomaly Types	Peak-pattern anomalies (e.g., SKP, PLP, FPT, FSK, PP)	Broad time series anomalies	Broad multivariate anomalies	Data drift and broad anomalies	Broad anomalies across tasks
Data Type Specialization	Specialized for environmental monitoring sensors	Broad applicability	Broad applicability	Broad applicability, with a focus on drift	Broad applicability
User Preference Integration	Yes (Model selection based on w)	No	No	No	No
Model Selection	Automated (Accuracy & efficiency)	Automated (Broad algorithms)	Automated (Broad algorithms)	Automated (Drift detection)	Automated (Meta-learning)
Synthetic Data Generation	Yes (Using TimeGAN for anomaly injection)	No	No	No	No
Targeted Anomalies	Yes (Specific to hydrology)	No (General anomalies)	No (General anomalies)	No (General anomalies)	No (General anomalies)
Deep Learning Models	Specialized (e.g., InceptionTime, MiniRocket)	Limited (General-purpose)	Limited (General-purpose)	Limited (General-purpose)	Limited (Adaptable, not hydrology-specific)
Framework Adaptability	High (Tailored for hydrological analysis)	Medium (General time series analysis)	Medium (Broad anomaly detection)	Medium (Online streaming data)	Medium (Adapts to new tasks)

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 (Lee et al., 2023).

captured using Turner Designs Cyclops-7 submersible sensors, known for their reliability and precision in environmental monitoring (see Fig. 1). The Turner Designs Cyclops-7 sensors are specifically designed for detecting fluorescence and turbidity in natural waters, making them ideal for assessing the stream fluxes of dissolved and particulate organic carbon in our study. Additionally, to account for variations in environmental conditions, the fDOM measurements were adjusted based on the turbidity values and water temperature. This data collection process forms the basis for our analysis, with further details on the dataset's preparation and utilization provided in Section 6.1.

3.2. Peak-pattern anomaly types

Anomalies in the fDOM and turbidity data were identified through visual examination and verified by a domain scientist. These identified anomalies were labeled and used to generate anomalies in the fully labeled synthetic peak pattern anomaly dataset. There are five types of such anomalies: skyrocketing peak (SKP), plummeting peak (PLP), flat plateau (FPT), flat sink (FSK), and phantom peak (PP). Fig. 2 shows examples of such peak patterns from the fDOM time series data. Skyrocketing peaks are characterized by a sharp upward spike or a narrow peak with a short base width, while a sharp downward spike characterizes plummeting peaks. These types of peaks may be caused by electronic sensor noise. Flat plateaus and flat sinks are characterized by a nearly constant signal amplitude at the top (plateau) and the bottom (sink), respectively, and may be caused by sediment deposits near or around the sensors. Flat sinks are only observed in fDOM data. Phantom peaks appear as normal peaks, but do not have a preceding stage rise that would trigger the peak. Non-hydrological events, such as animal activity in the water near the sensor may be the cause. To detect phantom peaks and plummeting peaks, it is necessary to consider the relationships between two data time series, while the other peak types can be identified using only one type of time series data.

4. Hydrology applications of the HF-PPAD

Our AutoML peak-pattern anomaly detection framework, HF-PPAD, aims to address a significant bottleneck in the field of hydrology — efficient removal of anomalous data from watershed time series data, which is necessary to analyze and model the data accurately. The HF-PPAD framework will improve the ability to find and access high-quality data and analysis codes, enabling scientists and educators to maximize the value of watershed data and produce transparent and reproducible research outcomes.

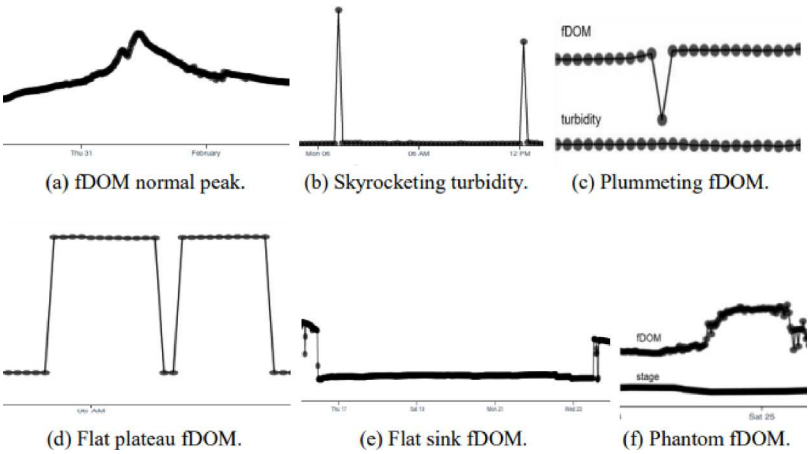


Fig. 2. Examples of anomalous peak-patterns types identified in fDOM time series data (Lee et al., 2023).

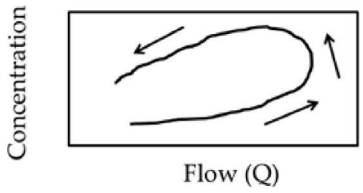


Fig. 3. Depiction of a C–Q hysteresis loop (Evans & Davies, 1998).

One specific application of the HF-PPAD framework is the analysis of concentration–discharge (C–Q) hysteresis, a phenomenon in which the concentration of a solute in a stream follows different trajectories on the rising and falling limbs of a storm or snowmelt discharge hydrograph. When the relationship between C and Q is nonlinear, this creates a loop on a plot of concentration against discharge (as shown in Fig. 3) and has long been of interest to hydrologists and biogeochemists seeking to interpret the size and direction of the loop over time as an indication of solute source and interactions with the watershed. The widespread deployment of in-stream sensors, measuring high-frequency chemistry at the same resolution as stream discharge has made it possible to construct finely-resolved hysteresis loops.

The testbed site is a small (41-ha) forested watershed in Vermont. At the outlet of the catchment, sensors are in place to measure stream water level, fluorescent dissolved organic matter (fDOM), turbidity, and water temperature. The water level is used to calculate stream discharge, fDOM is used as a proxy for dissolved organic carbon, and turbidity is a measure of particles in the water. fDOM is corrected for turbidity and water temperature following the method described in Downing et al. (2012). As is common at most sites, fDOM at W-9 generally increases with increasing discharge but with a delay such that it peaks after the stream discharge and has a long tail. This creates a counterclockwise hysteresis loop, with higher DOC concentrations at a given discharge on the falling limb compared with the same discharge on the rising limb (as shown in Fig. 4).

This application focuses on an fDOM time series that has already been corrected for turbidity and temperature using an automated process. However, the data still contain errors, often in the form of false peak patterns, that must be corrected before the time series can be used and accurately interpreted. The challenge is distinguishing normal peaks in fDOM (i.e., natural increases in fDOM with increases in flow) from false peaks caused by sensor malfunction, electrical surges, or other non-hydrological events such as a moose stirring up sediment in the gauge pool. Normal fDOM peaks should be accompanied by a rise in water level and usually a rise in turbidity. The HF-PPAD framework takes these clues into account and also is trained to differentiate peak types based on their shapes, with normal peaks generally having a broad base and an asymmetry skewed towards a long tail. Previous work on WTSD at SRRW (described in Lee et al. (2021) and Lee et al. (2023)) has identified normal and several anomalous peak types.

5. The AutoML pipeline of HF-PPAD framework

The fully automated pipeline of HF-PPAD framework is divided into two parts: one that automates creating a training set, and another that generates the best deep learning classifier through the tuning of architectural and training parameters of each model in the given pool. The generation of a model involves building and comparing different architectural instances of the model in conjunction with different training parameters. Additionally, the framework includes tools for generating time series data and injecting pattern anomalies into synthetic data. Fig. 5 shows an instance of the framework implemented in the current work.

5.1. Sub-models

Within this implementation, HF-PPAD encompasses a diverse array of sub-models, each selected from a pool of state-of-the-art deep learning models, to ensure a comprehensive approach. The sub-models are summarized below.

InceptionTime: A model inspired by the Inception network, known for its efficacy in handling time series data. It utilizes a combination of convolutional operations at different scales to capture time-dependent patterns effectively.

MiniRocket: Standing for 'Minimally Random Convolutional Kernel Transform', MiniRocket is highly efficient and scalable, using a diversified set of convolutional kernels to rapidly transform time series data for classification tasks.

ResNet: Short for Residual Network, ResNet is renowned for its deep architecture. It employs residual connections to facilitate the training of deeper networks by addressing issues like vanishing gradients, making it suitable for complex time series analysis.

TCN: Temporal Convolutional Networks (TCN) are specialized for sequence modeling, using causal convolutions to ensure that predictions for a specific time point are only dependent on past data, maintaining temporal coherence.

LSTM: Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) particularly adept at learning dependencies in sequence data. They are capable of capturing long-term dependencies, making them ideal for time series analysis where past information is crucial.

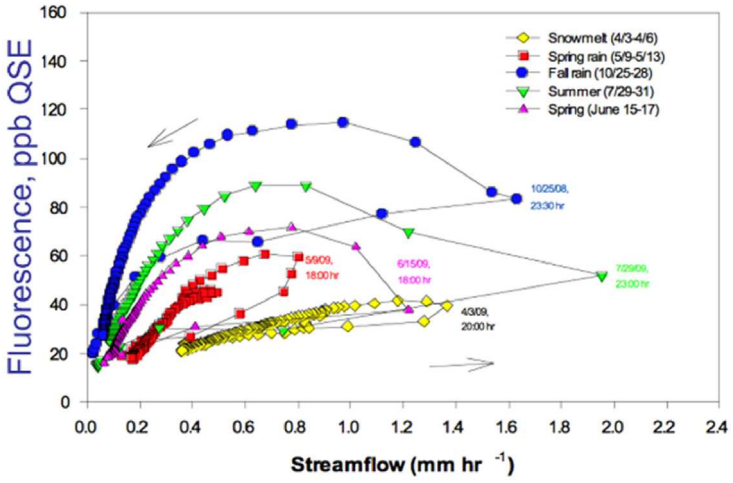


Fig. 4. Counterclockwise fDOM-Q hysteresis loops at Sleepers River, W-9 (Shanley, Sebestyen, McDonnell, McGlynn, & Dunne, 2015).

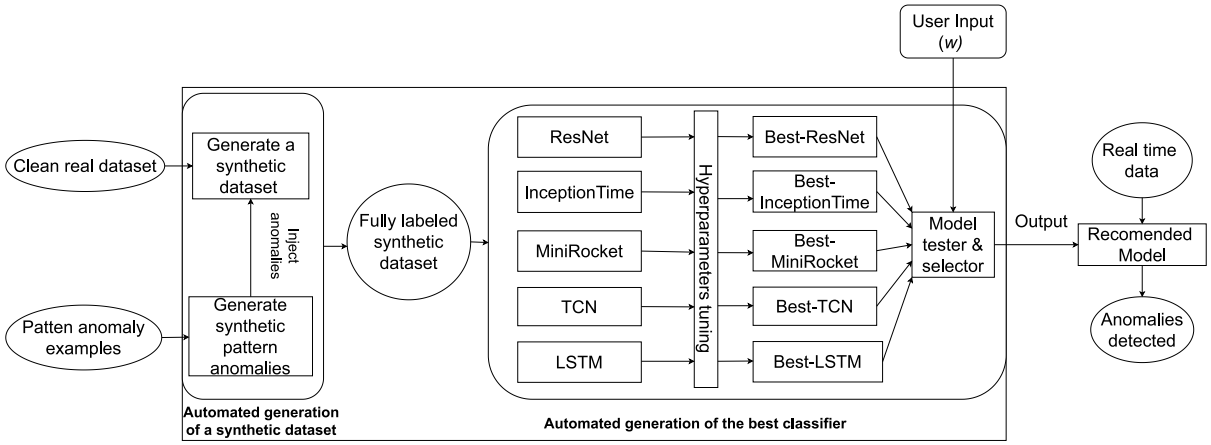


Fig. 5. The implemented HF-PPAD automated supervised machine learning framework.

5.2. Synthetic data generation

To generate synthetic watershed time series data (WTSD), we utilized TimeGAN, a state-of-the-art generative adversarial network (GAN) specifically designed for time series data synthesis. The initial step in this process involved selecting a subset of clean WTSD, typically encompassing one year of data, as a foundational dataset for generating synthetic data. The goal of creating a larger volume of synthetic data was to ensure the inclusion of a comprehensive and diverse set of peak pattern anomalies. This approach was crucial to provide enough training examples for the effective training and validation of our models, thereby improving the robustness and accuracy of anomaly detection.

TimeGAN was parametrized with several hyperparameters, carefully chosen to accurately replicate the characteristics of the WTSD. These parameters were configured as shown in Table 2.

The choice of Gated Recurrent Units (GRUs) as the module, coupled with a hidden dimension of 32 and four layers, was effective in capturing the temporal dynamics of the WTSD. To prevent any potential data leakage, the datasets used for generating synthetic data and anomalies were completely isolated from the datasets used for model testing The batch size of 256 and sequence length of 30 were selected to balance training efficiency with the model's ability to learn complex data patterns. The learning rate was set at 0.001 to ensure stable convergence during training. Additionally, the generator and discriminator activation functions, tanh and relu respectively, were chosen to improve the model's capacity to generate data closely resembling the original series.

Table 2
TimeGAN parameters for synthetic data generation.

Parameter	Value
Module	GRU
Hidden Dimension	32
Number of Layers	4
Iterations	5000
Batch Size	256
Sequence Length	30
Learning Rate	0.001
Generator Activation Function	tanh
Discriminator Activation Function	relu

Upon creating the synthetic dataset, we proceeded to augment it with synthetic anomalies. By generating altered versions of the identified peak pattern anomalies, we acquired a sufficient number of instances for each anomaly type, necessary for the effective training of our deep learning models. These synthetic anomalies were then randomly injected into the synthetic fDOM and turbidity data, ensuring a realistic representation of anomaly distribution.It is important to note that all testing datasets were strictly segregated from any data used in the synthetic generation and training processes, guaranteeing the integrity of our evaluation. The result was a comprehensively labeled training dataset, primed for deep learning classifier training. Fig. 6 illustrates the typical labeled peak-pattern anomalies injected into the generated synthetic time series data. This step was vital in

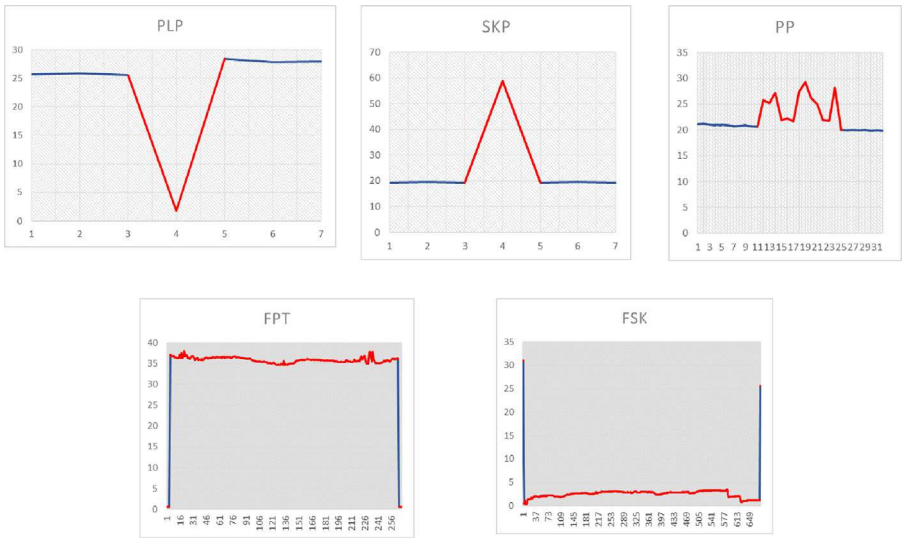


Fig. 6. Labeled anomalous peak patterns injected into synthetic time series data.

creating a dataset with diverse and authentic peak-pattern anomalies, facilitating the training of classifiers for effective anomaly detection in environmental time series data.

5.3. Generating the best deep learning classifier

The HF-PPAD framework employs a systematic process to determine the best deep learning classifier configuration from a set of potential models. This task is framed as optimizing a well-defined hyperparameter space, unique to each model type. The process involves automated tuning of model-specific architectural and training hyperparameters, utilizing advanced optimization algorithms. These algorithms explore various hyperparameter combinations to identify configurations that effectively balance accuracy and computational efficiency. This approach is particularly useful for domain experts in hydrology, who may lack in-depth machine learning expertise. By automating the complex task of hyperparameter tuning, the HF-PPAD framework provides an accessible pathway to deploy advanced deep learning models tailored to the specific needs of watershed time series data analysis.

5.3.1. Model instance search using hyperparameter optimization

Algorithm 1 outlines the AutoML algorithm of the HF-PPAD framework. This algorithm tunes each model in the given model pool one at a time using hyperparameter optimization techniques and outputs a model instance expected to achieve the top performance based on the evaluation results.

There are three aspects important to the efficacy of Algorithm 1: search space, search strategy, and evaluation strategy. Each is discussed below.

The *search space* is defined by a set of hyperparameters and their ranges. These ranges can be defined based on the specific needs and knowledge of the user. In our implementation of HF-PPAD, the hyperparameters are the machine learning models in the input pool, the architectural parameters pertaining to each model (see Table 3), and the training parameters that are common across all models (see Table 4). Overall, the search space allows for thoroughly exploring and optimizing various hyperparameters to identify the most suitable model instance and hyperparameter settings for a given data set.

The *search strategy* determines the process for iteratively selecting and evaluating combinations of hyperparameter values within the search space. The search strategy may be modified based on prior evaluations to improve future trials, or it may loop through all possible combinations within the search space. An effective search strategy can reduce the time required for the optimization process. For this work, we use Optuna, a tool for hyperparameter optimization that includes the four hyperparameter optimizers chosen in this work (i.e., random forest, Bayesian, Hyperband, and greedy). These optimizers are included as hyperparameters themselves in the search space, and on each trial, the AutoML algorithm selects the optimizer that provides the best result. The select optimizer then optimizes the architectural and training hyperparameters of the chosen model. The search time is directly proportional to the number of trials conducted. Increasing the number of trials can potentially improve the results but can also increase the tuning time.

The *evaluation strategy* is crucial, as it determines how the effectiveness of a model is evaluated with respect to its hyperparameters. The evaluation criteria, such as the validation performance and the total number of model parameters, are typically the same as those used in manual tuning. We also consider such factors as time/epoch, the number of parameters, and the memory usage for each model. By thoroughly evaluating the performance of each model and its corresponding hyperparameters, HF-PPAD can identify the most suitable model instance for a given data set.

Algorithm 1: AutoML algorithm of HF-PPAD against the WTSD.

Input : a pool of models $\{M_1, M_2, ..., M_n\}$;
synthetic watershed time series data (WTSD);
user's performance preference;
Output: the model instance showing the highest performance for the WTSD;

1 **for** each model M_i $(i = 1, 2, ..., n)$ in the pool **do**
2 Generate the best model instances $\hat{m}_i$ from the models in the pool that achieves the highest accuracy during training on synthetic WTSD by tuning M_i's architectural and training parameters through hyperparameter optimization;
3 Get the user's performance preference w and recommend the best model instance using Equation (1);
4 Test the recommended best model instance $Tr(\hat{m}_i)$ against the real test dataset to detect peak-pattern anomalies;
5 **end for**
6 Return the trained model instance that has the highest performance score in the result pool;

Table 3
Architectural hyperparameters of the individual deep learning model types used in HF-PPAD.

Hyperparameter	Domain
Number of layers	[18, 34, 50, 101, 152]
Number of filters	[16–1024]
Kernel size	[1, 3, 5, 7]
Stride	[1, 2]
Padding	[0, 1]
Pooling layer window size	[2 × 2,3 × 3]

(a) ResNet.

Hyperparameter	Domain
Number of Inception modules	[1–6]
Number of filters	[32–512]
Filter size	[3, 5, 7, 11]
Stride	[1, 2]
Pooling layer window size	[3–7]
Dropout rate	[0.1–0.5]

(b) InceptionTime.

Hyperparameter	Domain
Number of layers	[1–5]
Number of hidden units	[16–512]
Dropout rate	[0.1–0.5]
Recurrent dropout rate	[0.1–0.5]
Bidirectional	[yes, no]
Activation function	[Sigmoid, Tanh, ReLU]
Recurrent activation function	[Sigmoid, Tanh, ReLU]
Layer normalization	[yes, no]

(c) LSTM.

Hyperparameter	Domain
Number of layers	[2–100]
Kernel size	[1, 3, 5]
Dropout rate	[0.1–0.5]
Number of input channels	[1–64]
Number of filters	[32–1024]
Stride	[1, 2]
Dilation	[1–4]
Padding	[0, 1]

(d) TCN.

Hyperparameter	Domain
Number of random kernels	[100–5000]
Kernel sizes	[7–21]
Subsampling factor	[2–10]
Normalization	[true, false]
Number of random Fourier features	[1000–5000]

(e) MiniRocket.

Table 4
Training hyperparameters common to all the deep learning models in the pool.

Hyperparameter	Domain
Batch size	32, 64, 128, 256, 512
Optimizer	SGD, Adam
Learning rate	1e−6, 1e−5, 1e−4, 1e−3, 1e−2
Regularization	L1, L2, dropout

5.3.2. User preference-based best model instance selection

Consistent with recognized optimization practices in machine learning, which include aggregating objectives through linear or convex combinations, our framework facilitates model selection with a focus on user input. Following an optimization phase that identifies a set of top-performing models, the HF-PPAD framework introduces a step where a user-defined weight (w) is employed to guide the final model choice among these top models.This mechanism is crucial for aligning the selection with the user's specific requirements, as illustrated in the subsequent equation.

$$Q_{mi} = (1 - w) \times A_{mi} + w \times (1 - S_{mi}) \quad (1)$$

In this formulation, Q_{mi} balances anomaly detection accuracy (A_{mi}) with computational efficiency (S_{mi}), with w enabling users to adjust this balance. A higher value of w signals a user preference for accuracy, whereas a lower value indicates a preference for efficiency, thus allowing for an informed selection from the top candidates based on precise needs regarding accuracy and computational resource allocation. For practical interpretation, we convert the size S_{mi} into a more relatable metric like megabytes (MB) or gigabytes (GB), based on the data type used (typically float32). This step of incorporating user-defined weighting is designed to enhance the decision-making process within the automated machine learning context. It acknowledges the difficulty of predetermining an optimal balance between model performance and computational demand and provides a means for users to make decisions that reflect their operational constraints and priorities. Opting for user input to determine the final model selection from top performers is a deliberate choice to increase the framework's adaptability and user engagement. It bridges the optimization outcomes with user-specific application requirements, facilitating the use of advanced machine learning in diverse real-world scenarios without assuming prior optimization or machine learning expertise.

6. Evaluations

The HF-PPAD implementation performed on the WTSD used here has been evaluated thoroughly. There are three main questions answered through experiments:

- How similar is the synthetic time series dataset (with labeled peak-pattern anomalies injected) to the original real dataset from the WTSD? (See Section 6.3.)
- How well do the generated best individual deep learning models perform? (See Section 6.4.)
- How well does the autoML pipeline adapt to the user-specified preference between accuracy and computational cost to select the deep learning model that meets the preference best? (See Section 6.5.)

6.1. Datasets and data preparation

Our analysis focused on the fDOM and turbidity datasets collected over a period from October 1, 2012 to October 1, 2020. This extensive data range provided a diverse array for detailed exploration. In particular, data from October 1, 2016, to September 30, 2017, meticulously curated by domain experts, yielded about 35,000 data points for each of the fDOM and Turbidity datasets. These subsets formed the basis for our synthetic data generation. The synthetic data generation process, encompassing over 5000 epochs of training with TimeGAN, was pivotal to ensure a high-fidelity replication of the intrinsic features and patterns found in the original datasets.

In crafting our synthetic training dataset, we generated a total of 1,048,575 data points for each dataset, which included both clean data points and synthetic anomaly instances. For the fDOM dataset, 401,374 synthetic anomaly instances were introduced, amounting to 400–500 instances per anomaly type. The turbidity dataset similarly saw the introduction of 230,686 synthetic anomaly instances, translating to 500–600 instances per type. These numbers were specifically chosen to mirror the frequency and distribution of anomalies observed in the original datasets, ensuring a realistic and representative training set.

Table 5
Combined data statistics for synthetic training and original test datasets for fDOM and turbidity.

Synthetic training dataset				
Dataset	Anomaly types	Synthetic training Set Points	Injected anomalous points	Synthetic anomalies Per Type
fDOM	PLP, SKP, FPT, PP, FSK	1,048,575	401,374	400–500
Turbidity	SKP, FPT, PP	1,048,575	230,686	500–600
Original test dataset				
Dataset	Anomaly types	Total test points	Anomalous points in Test Set	Real anomalies Per Type
fDOM	PLP, SKP, FPT, PP, FSK	276,120	96,642	150–183
Turbidity	SKP, FPT, PP	276,120	60,500	233–260

This alignment with real-world data patterns was crucial for modeling the complex dynamics of peak-pattern anomalies in WTSD (as detailed in Table 5).

For model evaluation, we compiled test sets from the comprehensive dataset. Our initial research, documented in Lee et al. (2023), identified various anomaly types, including 'NAP' (Not A Peak). Subsequent analysis over an entire year extended our understanding of anomaly types. The resulting test sets consist of 276,120 fDOM data points containing 96,642 real anomaly points and 276,120 Turbidity data points containing 60,500 real anomaly points. The range of anomalies in the test sets is thoroughly detailed in the same Table 5. The fDOM dataset included anomalies such as PLP, SKP, FPT, PP, and FSK, each represented by 150–183 real instances. The Turbidity dataset featured anomalies types like SKP, FPT, and PP, with each type having 233–260 real instances. This extensive categorization of anomalies in the test sets significantly bolstered the robustness of our model evaluation, highlighting the intricate process of anomaly detection in environmental time series data.

This methodical approach, encompassing both the generation of advanced synthetic data via TimeGAN and the integration of real-world anomalies, ensured that our models were trained and validated under conditions that closely resembled the intricate scenarios of environmental time series data.

6.2. Experimental setup

The experimental setup for this study involved the use of an AutoML framework, HF-PPAD, to identify anomalous peak-pattern anomalies in WTSD.

Deep learning models. The deep learning models in the pool included InceptionTime, MiniRocket,ResNet,TCN and LSTM. Each model has its own search space for architectural hyperparameters and a common search space for training hyperparameters as discussed in Section 5.3.1. The tuning of these hyperparameters was carried out using Optuna, a hyperparameter optimization library. Four such optimizers, including random forest, Bayesian, Hyperband, and greedy search, were included in the search space to find the best model instance for each deep learning model. The hyperparameter optimization process for each deep learning model was run for 1000 trials with early stopping triggered when the validation loss did not improve for ten consecutive epochs. For validation, we used 70% of the training dataset, selected through shuffling. The resulting best model instances of the models were then trained for 50 epochs and tested against the WTSD test dataset using the user-provided performance objective (see Eq. (1)). The model instance that achieves the highest performance score in the test was then output.

Performance metrics. For the anomaly detection task, the performance achieved by a trained deep learning model comprises accuracy and computational cost. The accuracy used in this work are balanced accuracy (i.e., $\frac{1}{2}\left(\frac{TP}{TP+FN}+\frac{TN}{TN+FP}\right)$) and F-1 score (i.e., $2\cdot\frac{Precision\cdot Recall}{Precision+Recall}$). The computational costs are the time and memory consumed during model training. For simplicity, we use the number of model parameters as a proxy measure of computational cost, as both the training time and memory are proportional to it. We also report other parameters relevant to the model training, such as validation loss, epoch time, and the number of epochs.

Computing platform. All experiments were performed on Google Colab Pro platform, which provided access to a NVIDIA Tesla T4 GPU with 16 GB of memory and an Intel Xeon E5-2670 v3 CPU with 8 cores and 30 GB of memory. The programming language used was Python, with libraries including PyTorch and pandas.

6.3. Similarity of the synthetic dataset to the real dataset

As mentioned, the synthetic data points were generated using TimeGAN based on a clean dataset collected from the WTSD at SRRW. In order to evaluate the accuracy of the generated synthetic dataset, we selected two dominant variables, turbidity (for the x axis) and fDOM (for the y axis), from stage, turbidity, and fDOM through dimensionality reduction by PCA and by t-SNE, respectively, and generated clusters of the resulting data points in the 2D space of turbidity × fDOM. Fig. 7 shows the clusters of data points generated through PCA (left) and t-SNE (right). In both plots, the clusters of the original data points (blue) and the synthetic data points (red) are almost the same, which demonstrates the high similarity between the real and synthetic datasets.

To further verify the similarity, we aimed to assess the predictive accuracy of models trained on synthetic data compared to those trained on real data. This involved training an RNN regression model separately on real data and on synthetic data, and testing the two trained models against a separate real dataset. The RNN model consists of a single GRU (Gated Recurrent Unit) layer with 12 units and a dense output layer with six units and a Sigmoid activation function. The optimizer used is Adam, and the loss function is mean absolute error (MAE). This setup allowed us to directly evaluate the model's ability to predict future states based on the learned patterns from either real or synthetic data.

Table 6 summarizes the test accuracy (R-squared (R^2), mean absolute error (MAE), and mean squared error (MSE)) achieved by the two trained RNN models. The test accuracy of the model trained on the synthetic data was close to the test accuracy of the model trained on real data (within 4% for R^2, 2% for MAE, and 5% for MSE), confirming that the synthetic data generated by TimeGAN is a suitable substitute for real data in training machine learning models for the WTSD. This comparison underscores the synthetic data's efficacy in replicating the essential dynamics of the real dataset, thereby validating the use of TimeGAN-generated data for predictive modeling in water treatment studies.

6.4. Anomaly detection performances of the best instances of the models

HF-PPAD generated optimal model instances using synthetic datasets derived from the WTSD for training, and then tested these optimized model instances on the real dataset. Tables 7 and 8 summarize the performance results for each optimally trained model instance from

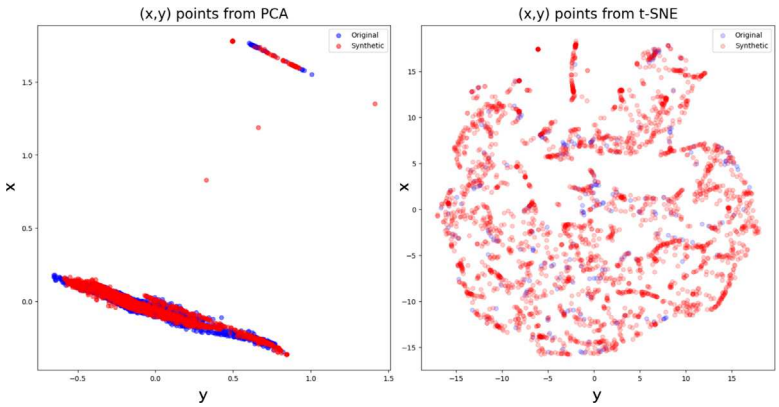


Fig. 7. Clusters of the synthetic and the original time series data points in a 2D turbidity × fDOM space generated by PCA (left) and t-SNE (right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Test accuracy of RNN regression models trained on synthetic dataset and real dataset and then tested on real dataset.

Training data	Test accuracy		
	R2	MAE	MSE
Synthetic	0.301858	0.016981	0.003859
Real	0.315577	0.016683	0.003672

Table 7
fDOM peak-pattern anomaly detection performance by the best trained model instance of each model in the HF-PPAD's model pool.

Model	Balanced accuracy	F-1 score	Number of parameters	Training time	Epoch time	Number of epochs
InceptionTime	97.3%	93.6%	1,817,888	350.5 s	7 s	50
ResNet	95.3%	90.1%	8,130,502	550.2 s	11 s	50
MiniRocket	93.4%	88.2%	89,974	150.6 s	3 s	50
LSTM	70.2%	64.7%	17 886	50.8 s	1 s	50
TCN	90.7%	85.9%	68,556	60.2 s	1.2 s	50

Table 8
Turbidity peak-pattern anomaly detection performance by the best trained model instance of each model in the HF-PPAD's model pool.

Model	Balanced accuracy	F-1 score	Number of parameters	Training time	Epoch time	Number of epochs
InceptionTime	95.3%	89.9%	4,082,884	467.2 s	9.34 s	50
ResNet	98.3%	94.6%	11,921,636	721.5 s	14.43 s	50
MiniRocket	91.6%	85.1%	118,974	349.7 s	6.94 s	50
LSTM	74.2%	67.7%	23 886	70.8 s	1.4 s	50
TCN	88.1%	81.9%	96,556	90.4 s	1.8 s	50

the pool of models. All five models achieved notable accuracy (ranging from 70.2% to 97.3% for balanced accuracy and 64.7% to 94.6% for F-1 score across fDOM and turbidity), which suggests the effective model generation capability of HF-PPAD. The computational costs varied across models, with some showing more significant differences than others. Notably, the optimally trained LSTM model instance, which recorded the lowest accuracy, also incurred the lowest computational cost. This observation highlights the trade-off and leads to the user-provided performance preference addressed below in Section 6.5.

To further examine model performance with a focus on the anomaly detection accuracy, we have created the confusion matrices shown in Fig. 8 for fDOM and Fig. 9 for turbidity. Overall, the detection accuracy for all peak-pattern anomaly types is notably high, which indicates the efficacy of the optimal model instance generation and training using the synthetic dataset. Particularly, the accuracy for the peak-pattern anomaly types FSK and FPT is 100% for all the optimal model instances; we attribute this accuracy to the long sequence of their anomaly instances that differentiate them from the other types of peak pattern anomalies. The accuracy for NAP is somewhat lower than for other anomaly types, as some instances are incorrectly classified as PP, PLP, or SKP peaks. It's important to note that NAP is not an anomalous peak type.

6.5. User input based best model instance selection

Recall that, the HF-PPAD approach recommends the best model instance for a dataset based on user preferences for accuracy and model size. Output quality was measured for the best trained model instance of each model using Eq. (1) and varying the weight parameter w from 0 to 1 at the increment of 0.2 for the fDOM and turbidity datasets. The results are shown as clustered bar charts in Fig. 10. The InceptionTime model instance had the highest accuracy for fDOM (0.973) at $w = 0$, whereas the TCN and MiniRocket model instances achieved the highest output quality (0.977 and 0.974, respectively) at $w = 0.8$. For turbidity, ResNet had the highest accuracy (0.983) at $w = 0$, while TCN and MiniRocket had the highest output quality (0.975 and 0.969, respectively) at $w = 0.8$. We can summarize that TCN and MiniRocket are recommended for users who prioritize accuracy and low computational cost, while InceptionTime and ResNet are best for users who prioritize high accuracy; and additionally that LSTM is recommended for users who prioritize low computational cost, despite its lower accuracy, as it has a smaller model size compared to the other models.

Fig. 11 shows a line graph of the output quality of the best model instance of each model as the user preference input w increases (at the increment of 0.1). It visualizes the trends of the output qualities changing between the different models. Specifically, it exhibits a decrease in the output quality of a model with a larger size as the w value increases. Notably, the MiniRocket and TCN models are competitive options for users who prioritize accuracy and low cost computational requirements. In contrast, the LSTM model only achieves higher output quality when w is 0 due to its smaller size. Overall, the figure highlights the varying output quality of the models and provides valuable insights into selecting the appropriate model based on user preferences.

7. Efficacy of the synthetic dataset

In this study, we examined the efficacy of using synthetic datasets in training machine learning models for anomaly detection in environmental time series data. This analysis was important in understanding the impact of synthetic data on model performance, especially given the scarcity of real-world training examples.

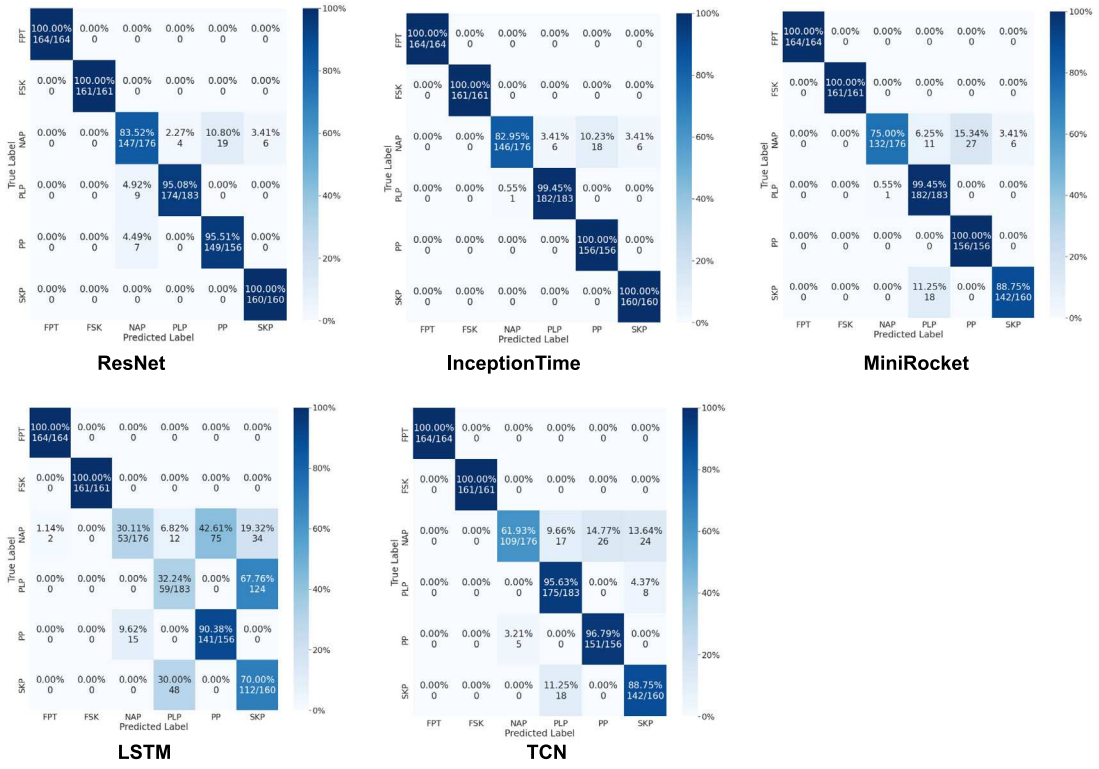


Fig. 8. Confusion matrix of fDOM peak-pattern anomaly detection accuracy by the best trained model instance of each model in the HF-PPAD's model pool.

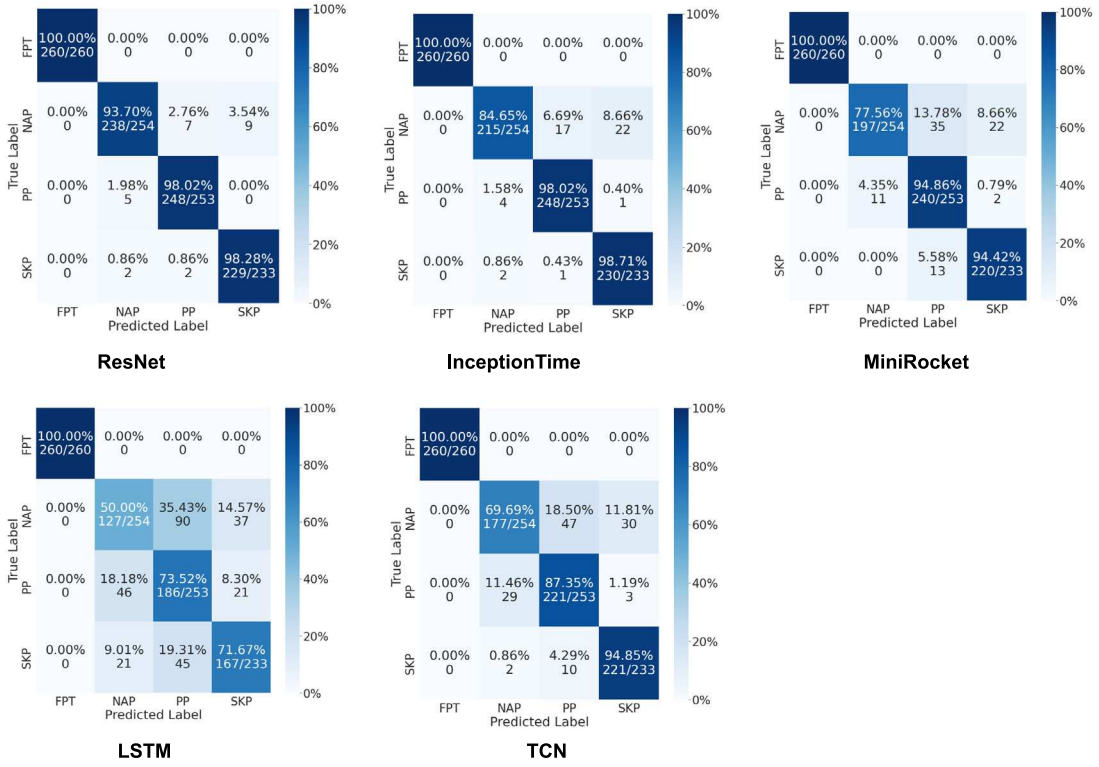
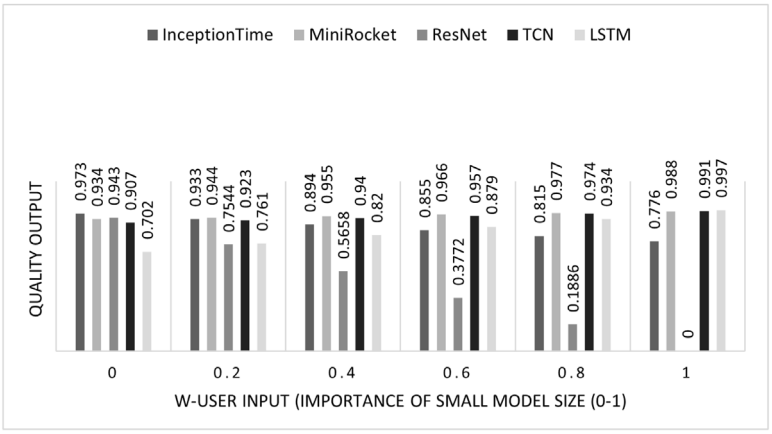
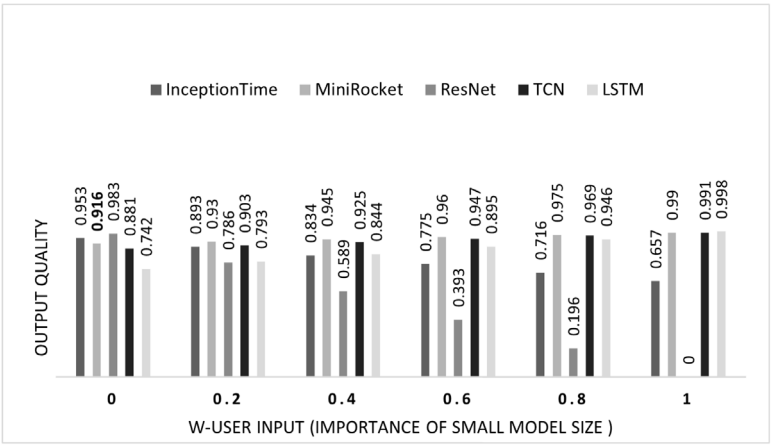


Fig. 9. Confusion matrix of turbidity peak-pattern anomaly detection accuracy by the best trained model instance of each model in the HF-PPAD's model pool.



(a) For fDOM WTSD.



(b) For turbidity WTSD.

Fig. 10. Comparison of the output quality achieved by the best model instance of each model for different values of the weight $w \in [0, 1]$; the weight indicates how much the user prefers small model size to high accuracy.

Table 9
Model performance comparison using manually labeled anomalies.

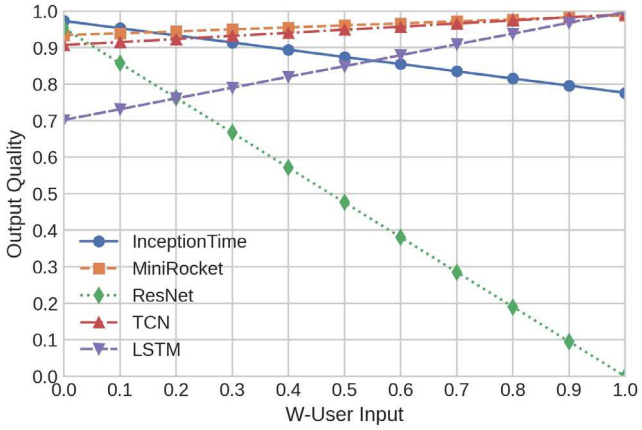
Model	fDOM Dataset		Turbidity Dataset	
	Balanced accuracy	F1 score	Balanced accuracy	F1 score
InceptionTime	50.3%	45.1%	47.6%	44.9%
ResNet	52.7%	47.5%	49.1%	47.3%
MiniRocket	49.8%	44.2%	45.8%	42.5%
LSTM	30.1%	25.6%	37.1%	33.8%
TCN	45.2%	40.8%	44.0%	40.9%

Initially, the models were trained using only manually labeled anomalies, which represented a limited range of training examples. This approach, while offering a valuable opportunity to evaluate the model's learning efficacy under constrained conditions, highlighted notable limitations. As shown in Table 9, the models exhibited a reduction in performance metrics, such as balanced accuracy and F1 scores, when trained exclusively on manually labeled data.

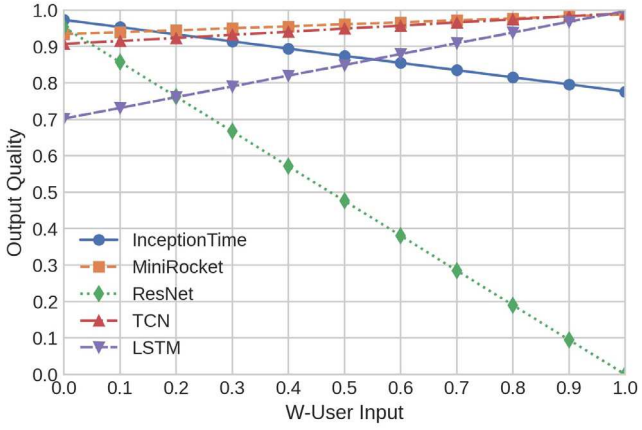
However, when the models were trained with synthetic datasets, there was a significant improvement in their performance. For the fDOM dataset, as illustrated in Fig. 12(a), and the turbidity dataset, shown in Fig. 12(b), the models demonstrated increased accuracy with the inclusion of synthetic data. The line graphs in these figures clearly show the progression of model accuracy as the percentage of synthetic data used in training increased from 0% to 100%.

The balanced accuracy for the fDOM dataset improved from 50.3% to 97.3% for InceptionTime and from 30.1% to 70.2% for LSTM, indicating a substantial enhancement in model performance with the inclusion of synthetic data. Similarly, for the turbidity dataset, the balanced accuracy saw an increase from 47.6% to 95.3% for InceptionTime and from 37.1% to 74.2% for LSTM.

These results underscore the importance of having extensive and varied training datasets in machine learning applications, particularly those involving complex environmental time series data. The introduction of synthetic data not only compensates for the scarcity of real-world labeled anomalies but also enriches the model's learning experience by introducing a broader spectrum of anomaly types and patterns. This study, adhering to rigorous standards of machine learning research, clearly demonstrates the necessity and effectiveness of comprehensive training datasets, including synthetic data, to enhance the learning capabilities of models and their subsequent performance in anomaly detection tasks.



(a) For fDOM WTSD.



(b) For turbidity WTSD.

Fig. 11. Changes of the output quality achieved by the best model instance of each model for the weight w increasing from 0 to 1.

8. Conclusion and future work

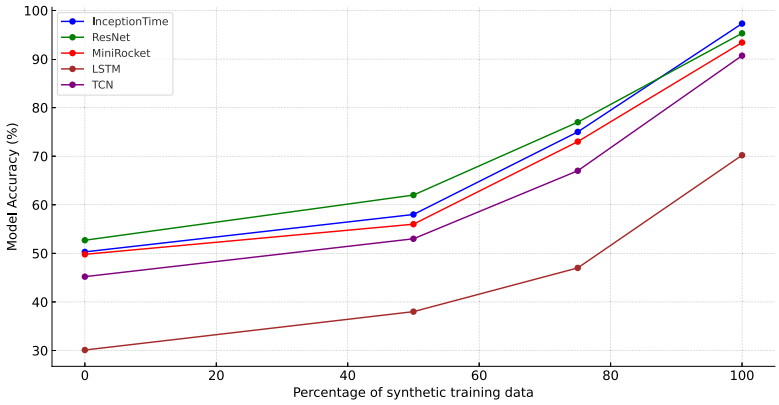
This paper introduced the HF-PPAD framework, a pioneering approach employing automated machine learning (AutoML) for detecting peak-pattern anomalies in watershed time series data (WTSD) from the northeast US critical zone. Our framework, which integrates a synthetic labeled dataset generator and an automated model instance generator, is tailored to assist hydrologists in identifying anomalous events in their data, such as peak-pattern anomalies in fDOM and turbidity, without requiring in-depth expertise in machine learning or anomaly detection algorithms.

While the framework demonstrates high performance in our experiments, particularly in its application to the WTSD, we acknowledge specific areas where further development and validation are essential. The validation of the framework, as it stands, is focused on specific datasets, and the nature of our anomaly identification process, which took almost a year to categorize peak pattern anomalies, highlights the challenge in generalizing this approach to other datasets. Future applications will necessitate a similar process of anomaly identification and categorization, underscoring the need for extensive testing to ensure the framework's adaptability to various datasets.

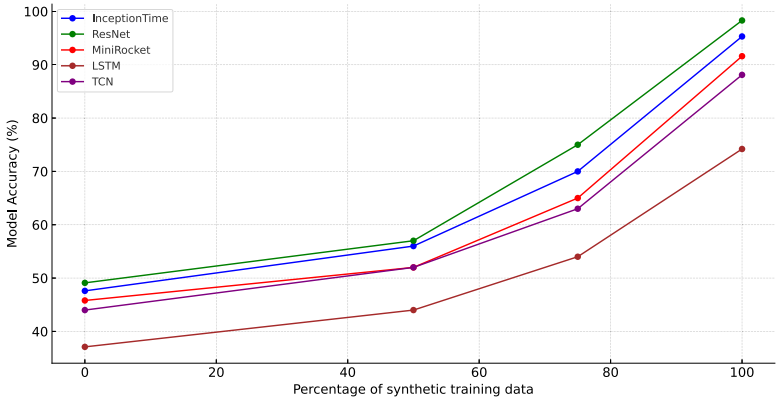
Moreover, the performance scalability of the framework with increasingly large and complex datasets remains an area for further exploration. The complexity of the proposed algorithms, particularly in managing the vastness of hyperparameter space and ensuring computational efficiency, poses significant challenges. As datasets grow in size and complexity, ensuring that the framework maintains its efficacy and efficiency becomes paramount. The balance between the accuracy of anomaly detection and computational resources, which is a cornerstone of our approach, needs to be constantly evaluated and optimized as we scale to larger datasets.

Our contribution is a notable first in utilizing AutoML for peak pattern anomaly detection in WTSD. By leveraging TimeGAN for synthetic dataset generation and incorporating a diverse pool of machine learning models (InceptionTime, ResNet, MiniRocket, TCN, and LSTM), we demonstrate the potential of AutoML for complex time series classification tasks in hydrology. However, we are cognizant that the current scope of our framework is primarily suited to the datasets and anomaly types we have studied. Expanding this scope to include additional machine learning models and a broader range of environmental data types, such as snow and air humidity, is an integral part of our future work. This expansion will not only test the framework's generalizability but also its applicability to other domains of anomalous events observed in water quality monitoring and flood forecasting.

In conclusion, the HF-PPAD framework stands as a significant step towards automated, efficient peak pattern anomaly detection in WTSD. By continuing to refine and expand its capabilities, and addressing the challenges highlighted in this study, we aim to establish HF-PPAD as an essential tool for hydrologists and stakeholders in water management. Our ongoing efforts will focus on enhancing the framework's generalizability, scalability, and computational efficiency, thereby broadening



(a) For fDOM WTSD.



(b) For turbidity WTSD.

Fig. 12. Model's accuracy for varying percentage of synthetic training data.

the horizons of AutoML applications in environmental science and beyond.

CRediT authorship contribution statement

Ijaz Ul Haq: Conceptualization, Methodology, Software development, Data Analysis, Writing – original draft, Visualization. **Byung Suk Lee:** Supervision, Project administration, Methodology, Writing – review & editing. **Donna M. Rizzo:** Data provision, Validation, Writing – review & editing. **Julia N. Perdrial:** Conceptualization, Writing – review & editing, Data analysis, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. This research was supported by the National Science Foundation, NSF, USA. All authors confirm that there are no conflicts of interest to declare.

Data availability

The authors do not have permission to share data.

Acknowledgments

This material is based upon work supported by the National Science Foundation, United States 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 D. James Shanley from the US Geological Survey (USGS) for offering the domain expertise that was crucial to identify the peak anomaly types that are of practical importance. The authors would also like to thank the anonymous reviewers, whose comments were invaluable to enhance the quality of the original manuscript.

References

Bahri, M., Salutari, F., Putina, A., et al. (2022). AutoML: state of the art with a focus on anomaly detection, challenges, and research directions. *International Journal of Data Science and Analytics*, *14*, 113–126. http://dx.doi.org/10.1007/s41060-022-00309-0.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3), 15:1–15:58. http://dx.doi.org/10.1145/1541880.1541882.

Cho, H., & Fryzlewicz, P. (2015). Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *77*(2), 475–507. http://dx.doi.org/10.1111/rssb.12079, arXiv:https://academic.oup.com/jrsssb/article-pdf/77/2/475/49214713/jrsssb_77_2_475.pdf.

Dempster, A., Schmidt, D. F., & Webb, G. I. (2021). MiniRocket: A very fast (almost) deterministic transform for time series classification. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 248–257). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3447548.3467231.

Deng, L., & Yu, D. (2014). Deep learning: Methods and applications. *Foundations and Trends® in Signal Processing*, *7*(3–4), 197–387. http://dx.doi.org/10.1561/2000000039.

Downing, J., Cole, J., Duarte, C., Middelburg, J., Melack, J., Prairie, Y., et al. (2012). Global abundance and size distribution of streams and rivers. *Inland Waters*, *2*(4), 229–236. http://dx.doi.org/10.5268/IW-2.4.502, https://www.tandfonline.com/doi/abs/10.5268/IW-2.4.502.

Dramsch, J. S. (2020). Chapter one - 70 years of machine learning in geoscience in review. In B. Moseley, & L. Krischer (Eds.), *Machine learning in geosciences Advances in geophysics*: *Advances in geophysics*: *vol. 61*, 1–55. http://dx.doi.org/10.1016/bs.agph.2020.08.002. https://www.sciencedirect.com/science/article/pii/S0065268720300054,

Enikeeva, F., & Harchaoui, Z. (2019). High-dimensional change-point detection under sparse alternatives. *The Annals of Statistics*, *47*(4), 2051–2079. http://dx.doi.org/10.1214/18-AOS1740.

Ergen, T., Mirza, A. H., & Kozat, S. S. (2017). Unsupervised and semi-supervised anomaly detection with LSTM neural networks. http://dx.doi.org/10.48550/arXiv.1710.09207, arXiv: Signal Processing (eess.SP); Machine Learning (cs.LG); Machine Learning (stat.ML), arXiv:1710.09207.

Evans, C., & Davies, T. D. (1998). Causes of concentration/discharge hysteresis and its potential as a tool for analysis of episode hydrochemistry. *Water Resources Research*, *34*(1), 129–137. http://dx.doi.org/10.1029/97WR01881.

Fearnhead, P., & Rigaill, G. (2019). Changepoint detection in the presence of outliers. *Journal of the American Statistical Association*, *114*(525), 169–183. http://dx.doi.org/10.1080/01621459.2017.1385466.

Feurer, M., & Hutter, F. (2019). Hyperparameter optimization. In F. Hutter, L. Kotthoff, & J. Vanschoren (Eds.), *Automated machine learning: methods, systems, challenges* (pp. 3–33). Cham: Springer International Publishing, http://dx.doi.org/10.1007/978-3-030-05318-5_1.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. In *Advances in neural information processing systems*: *vol. 28*, https://proceedings.neurips.cc/paper_files/paper/2015/file/11d0e6287202fced83f79975ec59a3a6-Paper.pdf.

Ghobadi, F., & Kang, D. (2023). Application of machine learning in water resources management: A systematic literature review. *Water*, *15*(4), http://dx.doi.org/10.3390/w15040620, https://www.mdpi.com/2073-4441/15/4/620.

Hamshaw, S. D., Denu, D., Holthuijzen, M., Wshah, S., & Rizzo, D. M. (2019). Automating the classification of hysteresis in event concentration-discharge relationships. In *SEDHYD 2019*. Reno, NV: SEDHYD, INC., https://www.sedhyd.org/2019/openconf/modules/request.php?module=oc_program&action=view.php&id=70&file=1/70.pdf.

Ho, L., & Goethals, P. (2022). Machine learning applications in river research: Trends, opportunities and challenges. *Methods in Ecology and Evolution*, *13*(11), 2603–2621. http://dx.doi.org/10.1111/2041-210X.13992, https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.13992.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Ismail Fawaz, H., Forestier, G., Weber, J., et al. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, *33*, 917–963. http://dx.doi.org/10.1007/s10618-019-00619-1.

Ismail Fawaz, H., Lucas, B., Forestier, G., et al. (2020). InceptionTime: Finding AlexNet for time series classification. *Data Mining and Knowledge Discovery*, *34*, 1936–1962. http://dx.doi.org/10.1007/s10618-020-00710-y.

Jin, H., Song, Q., & Hu, X. (2019). Auto-keras: An efficient neural architecture search system. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1946–1956). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3292500.3330648.

Jing, E., Zhang, H., Li, Z., Liu, Y., Ji, Z., & Ganchev, I. (2021). ECG heartbeat classification based on an improved ResNet-18 model. *Computational and Mathematical Methods in Medicine*, *2021*, Article 6649970. http://dx.doi.org/10.1155/2021/6649970.

Khan, M., Khan, S., & Alharbi, Y. (2020). Text mining challenges and applications, a comprehensive review. *International Journal of Computer Network and Information Security*, *20*, 138–148. http://dx.doi.org/10.22937/IJCSNS.2020.20.12.15.

Khan, Z. Y., Niu, Z., Nyamawe, A. S., & Haq, I. u. (2021). A deep hybrid model for recommendation by jointly leveraging ratings, reviews and metadata information. *Engineering Applications of Artificial Intelligence*, *97*, Article 104066. http://dx.doi.org/10.1016/j.engappai.2020.104066, https://www.sciencedirect.com/science/article/pii/S0952197620303316.

Kotthoff, L., Thornton, C., Hoos, H. H., Hutter, F., & Leyton-Brown, K. (2019). Auto-WEKA: Automatic model selection and hyperparameter optimization in WEKA. In *Automated machine learning: methods, systems, challenges* (pp. 81–95). Cham: Springer International Publishing, http://dx.doi.org/10.1007/978-3-030-05318-5_4.

Kulanuwat, L., Chantrapornchai, C., Maleewong, M., Wongchaisuwat, P., Wimala, S., Sarinnapakorn, K., et al. (2021). Anomaly detection using a sliding window technique and data imputation with machine learning for hydrological time series. *Water*, *13*(13), http://dx.doi.org/10.3390/w13131862, https://www.mdpi.com/2073-4441/13/13/1862.

Lai, K.-H., Zha, D., Wang, G., Xu, J., Zhao, Y., Kumar, D., et al. (2020). TODS: An automated time series outlier detection system. In *AAAI conference on artificial intelligence*. https://api.semanticscholar.org/CorpusID:221819569.

Lai, K. H., Zha, D., Wang, G., Xu, J., Zhao, Y., Kumar, D., et al. (2021). TODS: An automated time series outlier detection system. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 16060–16062). http://dx.doi.org/10.1609/aaai.v35i18.18012.

Lee, B. S., Kaufmann, J. C., Rizzo, D. M., & Haq, I. U. (2023). Peak anomaly detection from environmental sensor-generated watershed time series data. In J. A. Lossio-Ventura, J. Valverde-Rebaza, E. Díaz, & H. Alatrista-Salas (Eds.), *Information management and big data* (pp. 142–157). Cham: Springer Nature Switzerland.

Lee, B. S., Shanley, J., Fogg, Z., Rubin, J., Hamshaw, S., Rizzo, D., et al. (2021). *Automated cleaning of multiple time series data from the sleepers research river watershed.* New Orleans, LA: AGU Fall Meeting 2021 Abstracts, Abstract id. H45F-1245.

Li, H. Z., & Boulanger, P. (2020). A survey of heart anomaly detection using ambulatory electrocardiogram (ECG). *Sensors*, *20*(5), 1461. http://dx.doi.org/10.3390/s20051461.

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, *18*(1), 6765–6816.

Li, Y., Zha, D., Venugopal, P., Zou, N., & Hu, X. (2020). PyODDS: An end-to-end outlier detection system with automated machine learning. In *Proceedings of the web conference 2020* (pp. 153–157). http://dx.doi.org/10.1145/3366424.3383530.

Lin, Y., Lee, B. S., & Lustgarten, D. L. (2018). Continuous detection of abnormal heartbeats from ECG using online outlier detection. In *Symposium on information management and big data*. https://api.semanticscholar.org/CorpusID:53645008.

Pelletier, C., Webb, G. I., & Petitjean, F. (2019). Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing*, *11*(5), http://dx.doi.org/10.3390/rs11050523, https://www.mdpi.com/2072-4292/11/5/523.

Prasad, D. V. V., Venkataramana, L. Y., Kumar, P. S., Prasannamedha, G., Harshana, S., Srividya, S. J., et al. (2022). Analysis and prediction of water quality using deep learning and auto deep learning techniques. *Science of the Total Environment*, *821*, Article 153311. http://dx.doi.org/10.1016/j.scitotenv.2022.153311, https://www.sciencedirect.com/science/article/pii/S004896972200403X.

Qin, Y., & Lou, Y. (2019). Hydrological time series anomaly pattern detection based on isolation forest. In *2019 IEEE 3rd information technology, networking, electronic and automation control conference* (pp. 1706–1710). http://dx.doi.org/10.1109/ITNEC.2019.8729405.

Ryzhikov, A., Borisyak, M., Ustyuzhanin, A., & Derkach, D. (2021). NFAD: Fixing anomaly detection using normalizing flows. *PeerJ Computer Science*, *7*, Article e757. http://dx.doi.org/10.7717/peerj-cs.757.

Schmidl, S., Wenig, P., & Papenbrock, T. (2022). Anomaly detection in time series: A comprehensive evaluation. *Proceedings of the VLDB Endowment*, *15*(9), 1779–1797. http://dx.doi.org/10.14778/3538598.3538602.

Schmidt, L., Heße, F., Attinger, S., & Kumar, R. (2020). Challenges in applying machine learning models for hydrological inference: A case study for flooding events across Germany. *Water Resources Research*, *56*(5), Article e2019WR025924. http://dx.doi.org/10.1029/2019WR025924, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019WR025924.

Senagi, K. M. (2019). *Random forest hyperparameter optimization, GPU parallelization and applications to soil analysis for optimal crop production* (Ph.D. thesis). Thèse de doctorat dirigée par Jouandeau, Nicolas Informatique Paris 8 2019. http://www.theses.fr/2019PA080086.

Shanley, J. B., Sebestyen, S. D., McDonnell, J. J., McGlynn, B. L., & Dunne, T. (2015). Water's way at sleepers river watershed – Revisiting flow generation in a post-glacial landscape, vermont USA. *Hydrological Processes*, *29*(16), 3447–3459. http://dx.doi.org/10.1002/hyp.10377, https://onlinelibrary.wiley.com/doi/abs/10.1002/hyp.10377.

Shen, C., Chen, X., & Laloy, E. (2021). Editorial: Broadening the use of machine learning in hydrology. *Frontiers in Water*, *3*, http://dx.doi.org/10.3389/frwa.2021.681023, https://www.frontiersin.org/articles/10.3389/frwa.2021.681023.

Sit, M., Demiray, B. Z., Xiang, Z., Ewing, G. J., Sermet, Y., & Demir, I. (2020). A comprehensive review of deep learning applications in hydrology and water resources. *Water Science and Technology*, *82*(12), 2635–2670. http://dx.doi.org/10.2166/wst.2020.369, arXiv:https://iwaponline.com/wst/article-pdf/82/12/2635/802982/wst082122635.pdf.

Sun, J., Lou, Y., & Ye, F. (2017). Research on anomaly pattern detection in hydrological time series. In *2017 14th web information systems and applications conference* (pp. 38–43). http://dx.doi.org/10.1109/WISA.2017.73.

Tveten, M., Eckley, I. A., & Fearnhead, P. (2022). Scalable change-point and anomaly detection in cross-correlated data with an application to condition monitoring. *The Annals of Applied Statistics*, *16*(2), 721–743. http://dx.doi.org/10.1214/21-AOAS1508.

Vaughan, M. C. H., Bowden, W. B., Shanley, J. B., Vermilyea, A., Sleeper, R., Gold, A. J., et al. (2017). High-frequency dissolved organic carbon and nitrate measurements reveal differences in storm hysteresis and loading in relation to land cover and seasonality. *Water Resources Research*, *53*(7), 5345–5363. http://dx.doi.org/10.1002/2017WR020491.

Wu, Y., Xi, X., & He, J. (2022). AFGSL: Automatic feature generation based on graph structure learning. *Knowledge-Based Systems*, *238*, Article 107835. http://dx.doi.org/10.1016/j.knosys.2021.107835.

Yao, Q., Wang, M., Chen, Y., Dai, W., Li, Y.-F., Tu, W.-W., et al. (2018). Taking human out of learning applications: A survey on automated machine learning. arXiv:1810.13306.

Yoon, J., Jarrett, D., & van der Schaar, M. (2019). Time-series generative adversarial networks. In *Advances in neural information processing systems*: *vol. 32*, https://proceedings.neurips.cc/paper_files/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf.

Yu, Y., Wan, D., Zhao, Q., & Liu, H. (2020). Detecting pattern anomalies in hydrological time series with weighted probabilistic suffix trees. *Water*, *12*(5), http://dx.doi.org/10.3390/w12051464, https://www.mdpi.com/2073-4441/12/5/1464.

Zha, D., Lai, K.-H., Wan, M., & Hu, X. B. (2020). Meta-AAD: Active anomaly detection with deep reinforcement learning. In *2020 IEEE international conference on data mining* (pp. 771–780). https://api.semanticscholar.org/CorpusID:221738930.

Zhao, Y., Nasrullah, Z., & Li, Z. (2019). PyOD: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, *20*(96), 1–7, http://jmlr.org/papers/v20/19-011.html.

Zhu, Y., Wu, Q., & Liu, J. (2023). A comparative study of contrastive learning-based few-shot unsupervised algorithms for efficient deep learning. *Journal of Physics: Conference Series*, *2560*(1), Article 012048. http://dx.doi.org/10.1088/1742-6596/2560/1/012048.