

# 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<sup>a</sup>, Byung Suk Lee<sup>a,\*</sup>, Donna M. Rizzo<sup>b</sup>, Julia N Perdril<sup>c</sup>

<sup>a</sup>*Department of Computer Science, University of Vermont, 05405, Burlington, VT, U.S.A*

<sup>b</sup>*Department of Civil and Environmental Engineering, University of Vermont, 05405, Burlington, VT, U.S.A*

<sup>c</sup>*Department of Geography and Geosciences, University of Vermont, 05405, Burlington, VT, U.S.A*

---

## 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

---

\*This document is the results of the research project funded by the National Science Foundation.

\*Corresponding author

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.

*Keywords:*

automated machine learning, anomaly detection, time series data, watershed, sensor-generated data, hyperparameter optimization, deep learning models

---

## 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 pro-

cess. Detecting anomalies in WTSD is challenging 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 and Davies (1998), Hamshaw et al. (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 efficient solution for detecting pattern anomalies in WTSD, making it accessible and convenient for domain scientists. It needs thorough understanding of anomaly detection algorithms for users to choose the right one, which often requires a strong background in generative models and statistical assumptions. Properly setting the parameters for these algorithms often requires detailed understanding of their inner workings. Most domain scientists (often hydrologists and biogeochemists in this case) are lacking such background, and HF-PPAD is stepping in to help. HF-PPAD utilizes *supervised* deep learning models to deliver more accurate 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 exceptional results in various machine-learning tasks (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 et al. (2021)). ResNet is a widely recognized model known for its accuracy and has been adapted for time series data analysis (Jing et al. (2021)). InceptionTime, on the other hand, is specifically designed for analyzing time series data (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 et al. (2019)). Additionally, our choice of LSTM was based on its proven effectiveness in a wide range of time series applications (Hochreiter and Schmidhuber (1997)). These models can be configured in a variety of ways, with ResNet, InceptionTime, and LSTM being more powerful, 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 et al. (2019)) to automatically generate a large amount of time series data containing labeled peak pattern anomalies similar to the original peak-pattern anomalies; this eliminates 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 and 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. An end-to-end automated peak anomalous pattern detection framework for watershed time series data.
2. The use of TimeGAN to generate labeled synthetic watershed time

series data and peak pattern anomalies.

3. An automated generation (i.e., design and selection) of 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.

In the remainder of the paper, Section 2 discusses related work, Section 3 discusses the application of HF-PPAD. Section 4 provides an overview of watershed data and the different peak-pattern anomaly types. Section 5 outlines the AutoML pipeline of the HF-PPAD framework, including its data preparation and model selection steps. Section 6 presents the results of our experiments. Finally, Section 7 concludes the paper and discusses avenues for future research.

## 2. Related Work

### 2.1. Peak anomaly detection

Anomaly detection methods can be used to identify different types of anomalies, including point anomalies, pattern anomalies, and system anomalies (Lai et al. (2021); Chandola et al. (2009)). A point anomaly refers to a single sample in a time series, whereas a pattern anomaly is identified by a sequence of samples that exhibit a certain characteristic or behavior (e.g., trend, change). A system anomaly refers to a group of sequences (e.g., sets of time series patterns) in which one or more systems are in an abnormal state. Most existing work on anomaly detection has focused on identifying point anomalies (Cho and Fryzlewicz (2015); Enikeeva and Harchaoui (2019); Fearnhead and Rigaiil (2019); Fryzlewicz (2014); Tveten et al. (2022)). Pang et al. (2021) noted that methods for detecting point anomalies cannot be applied to “group anomalies” with distinct characteristics. The reference to group anomalies in our work is also the same as pattern anomalies.

The peak anomaly in our watershed data is a pattern anomaly that is identified by the shape of the time series sample sequences. There have been a few efforts to detect pattern anomalies in hydrological watershed sensor-generated time series data, but these efforts have primarily focused on detecting deviations from patterns (Yu et al. (2020), Sun et al. (2017) and Qin and Lou (2019)). The peak anomalies that we are interested in are different from the pattern anomalies detected by these algorithms. We have found that there is more relevant work on detecting peak anomalies in time series data from other domains, such as Electrocardiogram (ECG)

anomaly detection (Lin et al. (2019), Li and Boulanger (2020)). These ECG datasets are annotated with codes indicating whether segments are normal or abnormal at each R peak location.

### *2.2. Automated machine learning in hydrology*

Automated machine learning (AutoML) has emerged as a promising solution for enhancing anomaly detection in hydrology. Despite the application of machine learning in hydrology for over 70 years (Dramschi (2020)), selecting the most suitable model for a given problem remains a challenge. In recent years, the focus of machine learning in hydrology has shifted toward model validation, applied statistics, and subject matter expertise.

Automated machine learning (AutoML), a field that automates the processes and tasks involved in machine learning problems (Wu et al. (2022), Yao et al. (2018)), has the potential to enhance anomaly detection methods. Although still in its early stages, a few proposed methods use AutoML for anomaly detection (Li et al. (2021), Neutatz et al. (2022)). Most techniques are designed to solve a specific problem or work with certain data constraints.

Existing AutoML tools such as Auto-WEKA (Kotthoff et al. (2019)) and Auto-Sklearn (Feurer et al. (2015)) lack the more modern automated approaches for deep learning models. Auro-Keras (Jin et al. (2019)), an open-source library, optimizes deep neural networks for text and image data only and is not specifically designed for time series classification tasks. Also, these tools provide a single optimizer for hyperparameter optimization. Our AutoML pipeline extends the optimization framework to include deep learning model architectures, training hyperparameters, as well as the optimization strategies (e.g., random forest, Bayesian, Hyperband, and greedy search algorithms) to select the best optimizer for generating optimal model instances. Furthermore, our framework represents a novel application of AutoML approaches to deep learning time series classifiers for detecting peak pattern anomalies in WTSDs. This approach transforms the anomaly detection task into a supervised classification task.

### *2.3. Unsupervised/semi-supervised versus supervised anomaly detection*

Deep learning models, including supervised, semi-supervised, and unsupervised, have become increasingly popular in a wide range of domains due to their ability to process complex data and learn patterns (Deng and Yu. (2021), Khan et al. (2021), Haq et al. (2021), Matar et al. (2021)). However, when it comes to anomaly detection in time series data, *unsupervised* or

*semi-supervised* learning methods are often preferred, as obtaining anomaly labels can be challenging or impractical (e.g., Bahri et al. (2022); Schmidl et al. (2022)). These methods often employ shallow learning techniques like clustering or deep learning such as LSTM-based regression, autoencoders, and generative adversarial networks (GANs). The accuracy of anomaly detection may be compromised due to the lack of human input and oversight in learning what constitutes an anomaly; and unsupervised or semi-supervised learning generally requires more computational resources (in terms of training time and memory consumption) than supervised learning (Bahri et al. (2022)).

AutoML has also been applied in *unsupervised* tasks, such as clustering data and predicting clusters for new observations (Koren et al. (2022)), discovering optimized hyper-parameters of a model (Bahri et al. (2022)), selecting anomaly detection models (Kotlar et al. (2021)), detecting outliers in time series data (Kancharla and Raghu Kishore (2022), Shende et al. (2022), Xing et al. (2022), Xiao et al. (2021)), generating labeled data (Chatterjee et al. (2022)) and finding anomalies in images (Sawaki et al. (2019)). There are several other existing AutoML anomaly detection frameworks, such as PyOD (Zhao and Nasrullah (2019)), PyODDS (Li et al. (2020)), MetaAAD (Zha et al. (2020)), and TODS (Lai et al. (2021)), that are designed to identify anomalies in data. These frameworks are all *unsupervised* and primarily target solving point anomaly or change-point detection problems, rather than the peak-pattern anomaly detection problem that our framework aims to address.

There are *supervised* learning methods developed for point anomaly detection from time series data, such as those in Ryzhikov et al. (2020) and Li et al. (2017). However, none is designed to detect peak-pattern anomalies. In addition, a survey by Schmidl et al. (2022) found that existing supervised methods for anomaly detection from time series data are limited to binary classification of each time series data point into normal and abnormal. Our work, in contrast, performs multi-class classification to detect multiple types of peak-pattern anomaly.

### 3. 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 neces-

sary 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.

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 Figure 1) 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.

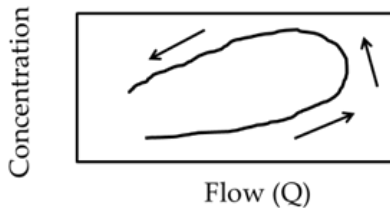


Figure 1: Depiction of a C-Q hysteresis loop (source: Evans and Davies (1998)).

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 Figure 2).

This application focuses on an FDOM time series that has already been



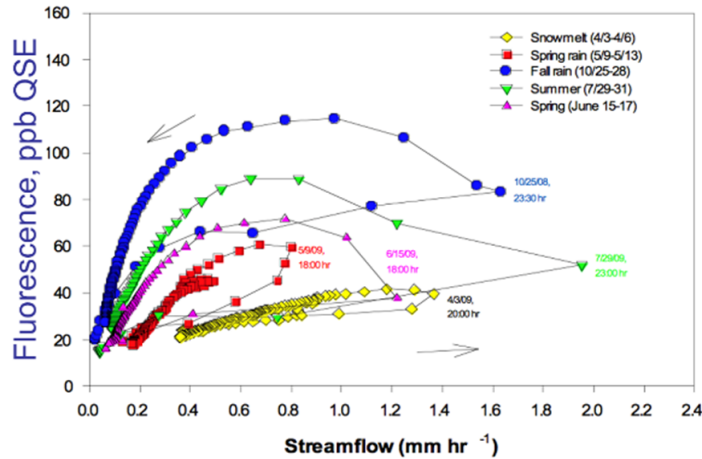


Figure 2: Counterclockwise FDOM-Q hysteresis loops at Sleepers River, W-9 (from Shanley et al. (2015)).

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. 2022) has identified normal and several anomalous peak types.

## 4. Watershed Data and Peak-Pattern Anomaly Types

### 4.1. Watershed time series data

Sensor data were collected from the study watershed over a period six years and four months (from October 1, 2012 to January 1, 2019). The measurements of stream stage, turbidity and FDOM were taken at 5-minute intervals for stream stage and 15 minutes for turbidity and FDOM. The measurements were taken using Turner Designs Cyclops sensors (see Figure 3),

and used to estimate the stream fluxes of dissolved and particulate organic carbon. The FDOM measurements were adjusted based on the turbidity values and the water temperature. The stage data included 231,465 points, and the turbidity and FDOM data each included 229,620 points.



Figure 3: Turbidity/FDOM sensor mounted on a board immersed in the water. The image in the corner is a Turner Designs Cyclops-7 submersible sensor.

#### *4.2. 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). Figure 4 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.

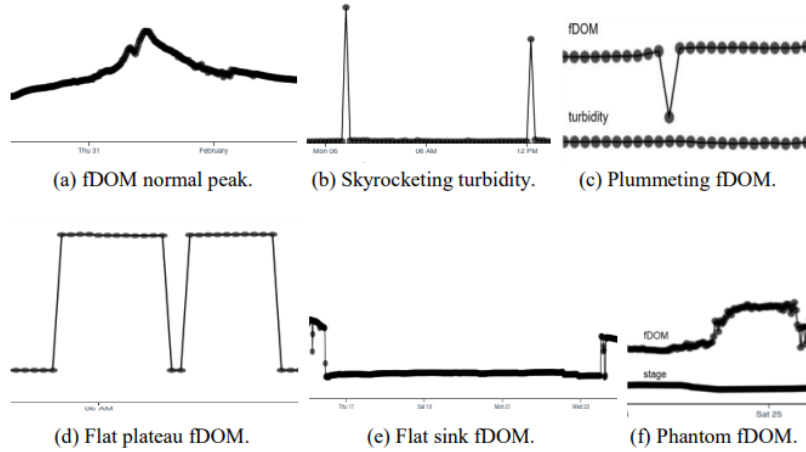


Figure 4: Examples of anomalous peak-patterns types identified in fDOM time series data.

## 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. Figure 5 shows an instance of the framework implemented in the current work. In this implementation, HF-PPAD includes a range of sub-models drawn from a pool of state-of-the-art deep learning models, such as InceptionTime, MiniRocket, ResNet, TCN and LSTM as well as tools for generating time series data, injecting pattern anomalies into synthetic data, and tuning hyperparameters.

### 5.1. Synthetic data generation

To generate synthetic watershed time series data (WTSD), we utilize the state-of-the-art time series generator TimeGAN, which uses a generative

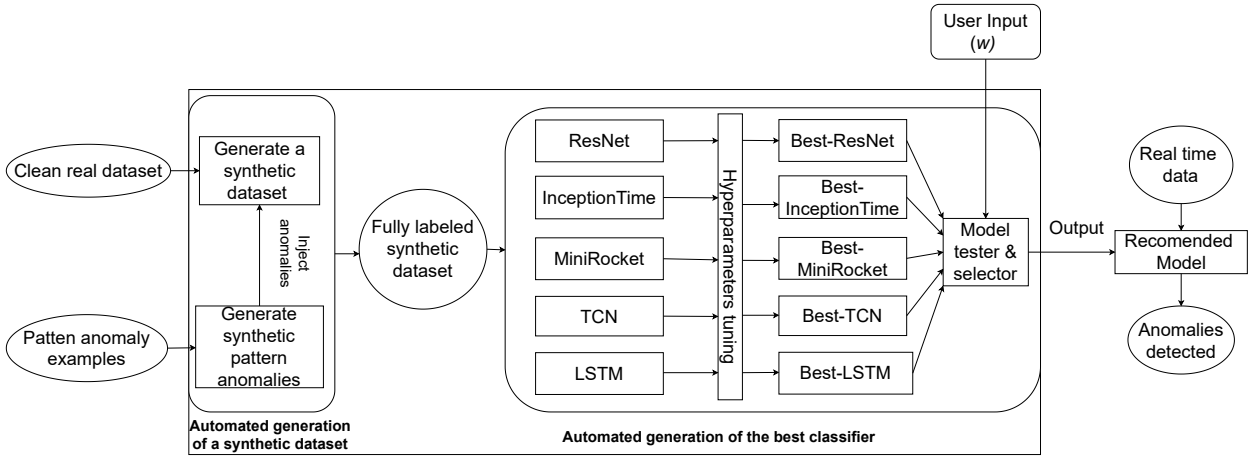


Figure 5: The implemented HF-PPAD automated supervised machine learning framework.

adversarial network (GAN) to output data that is nearly identical to the input data. We begin by obtaining a small portion of clean WTSD, such as clean data of one year, and use it to generate a large amount of synthetic data with TimeGAN.

To create a labeled dataset for supervised learning, we augment and inject anomalies into synthetic data using a small number of ground truth labels. This process enables us to create a sufficiently large training dataset with minimal manual labeling while also addressing data sparsity and skewness issues that are common in watershed time series data.

To generate synthetic anomalies, we, again, utilize the state-of-the-art time series generator TimeGAN. By generating multiple altered versions of the identified peak pattern anomalies, we have obtained a sufficient number of instances of each anomaly type to train the deep learning models. These synthetic anomalies are then injected at random positions within the synthetic FDOM and turbidity time series data generated by TimeGAN to mimic the random occurrence of anomalies in real data. This results in a fully prepared and labeled training dataset for the deep learning classifiers. The importance of this step lies in creating a multi-class labeled peak pattern anomaly dataset suitable for training deep learning classifiers. Figure 6 shows the typical labeled peak-patterns anomalies injected into the generated synthetic time series data.

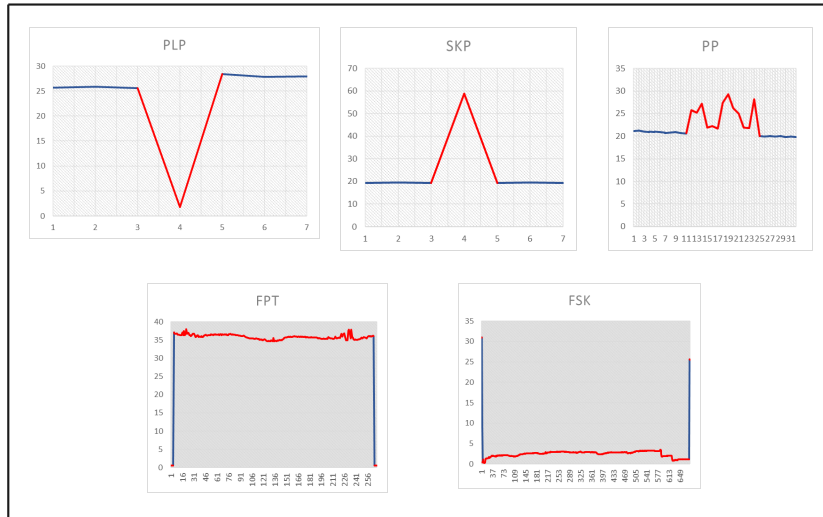


Figure 6: Labeled anomalous peak patterns injected into synthetic time series data.

## 5.2. Generating the best deep learning classifier

HF-PPAD handles the best model generation problem as an optimal search problem in a parameter space pertaining to the models. Each model has its own search space that includes a range of individual architectural and training hyperparameters to choose from. These hyperparameters are automatically tuned using optimizers to find the best model instance from a pool of select models. This automated process is particularly helpful for hydrologists who may not have adequate expertise in machine learning.

### 5.2.1. 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 1), and the training parameters that are

---

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

---

**Input** : a pool of models  $\{M_1, M_2, \dots, M_n\}$ ;  
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  $M_i$  ( $i = 1, 2, \dots, n$ ) 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;

---

common across all models (see Table 2). 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 improve the results but can also increase the tuning

Hyperparameter	Domain	Hyperparameter	Domain
Number of layers	[18, 34, 50, 101, 152]	Number of Inception modules	[1 – 6]
Number of filters	[16 – 1024]	Number of filters	[32 – 512]
Kernel size	[1, 3, 5, 7]	Filter size	[3, 5, 7, 11]
Stride	[1, 2]	Stride	[1, 2]
Padding	[0, 1]	Pooling layer window size	[3 – 7]
Pooling layer window size	[2×2, 3×3]	Dropout rate	[0.1 – 0.5]

(a) ResNet.

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]

(c) MiniRocket.

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]

(d) 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]

(e) TCN.

Table 1: Architectural hyperparameters of the individual deep learning model types used in HF-PPAD.

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.

### 5.2.2. User preference-based best model instance selection

Our automated peak-pattern anomaly detection framework HF-PPAD assists users in identifying the most appropriate machine learning model for

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

Table 2: Training hyperparameters common to all the deep learning models in the pool.

their WTSD. The algorithm conducts exhaustive tuning of architectural and training hyperparameters to determine the optimal instance of each model. The effectiveness of each model instance is then evaluated using Equation 1, which takes into account both the accuracy achieved and the computational cost incurred during the tuning process. The user is also asked to specify a weight indicating the relative importance of lower computational cost (e.g., training time and memory usage) compared to higher accuracy. This weight is linearly related to the computational cost and allows for personalized model instance recommendations based on the user’s specific needs and preferences. By doing so, it helps users to make informed decisions about the best model instance for their specific data set, considering both performance and computational cost.

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

where  $m_i$  ( $i = 1, 2, \dots, n$ ) is an instance of a model  $M_i$  in the pool;  $Q_{m_i}$  is the output quality achieved using the model instance  $m_i$ ;  $A_{m_i}$  is the accuracy achieved using the model instance  $m_i$ ;  $S_{m_i}$  is the size of  $m_i$  normalized by the maximum possible size of all instances of  $M_i$  and  $w$  is the user-provided weight of a smaller model size (i.e.,  $(1 - S_{m_i})$ ) over higher accuracy (i.e.,  $A_{m_i}$ ). The size of a deep learning model can be determined by looking at the number of parameters. To determine the size of a deep learning model, we convert the total number of parameters to a more readable format, such as megabytes (MB) or gigabytes (GB), by dividing by the number of bytes per parameter (usually 4 for float32 data type).

## 6. Experiments

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 6.2.)
- How well do the generated best individual deep learning models perform? (See Section 6.3.)
- 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 6.4.)

### 6.1. Setup

*Datasets.* One year (from October 1, 2016 to September 30, 2017) worth of clean FDOM and turbidity data was used in all the experiments. This dataset has 105,120 points. They were passed to TimeGAN, the machine learning algorithm selected for generating synthetic labeled data, to create a dataset of 1,048,575 time series samples; the dataset was then split into 70% training and 30% validation datasets. TimeGAN was then trained for 5,000 epochs, as recommended by Yoon et al. (2019), to ensure that it captured the main features and patterns of the real data. Subsequent to the generation of the clean synthetic data, TimeGAN was used to generate synthetic instances of anomalous peak patterns (see Figure 6). Synthetic versions of 400 to 500 anomalous peak patterns were created for each type of anomaly and randomly injected into the synthetic dataset. The resulting dataset containing clean time series samples interspersed with anomalous peak patterns was used to train and validate the deep learning models in the pool. The trained models were then tested on real data containing real peak pattern anomalies.

*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 5.2.1. 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 1,000 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 Equation 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 16GB of memory and an Intel Xeon E5-2670 v3 CPU with 8 cores and 30GB of memory. The programming language used was Python, with libraries including PyTorch and pandas.

## 6.2. 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  $\times$  FDOM. Figure 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 trained an RNN regression model separately on real data and on synthetic data, and tested 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

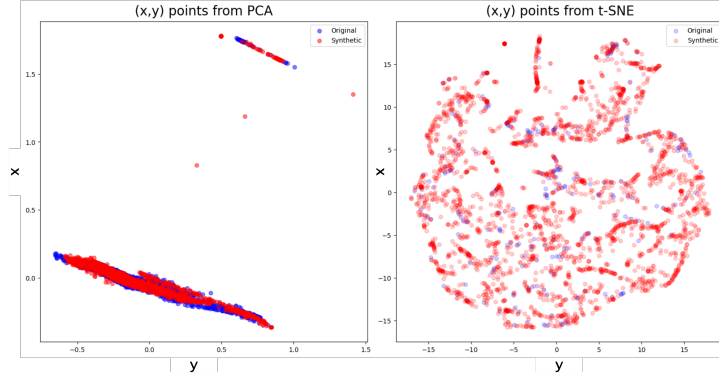


Figure 7: Clusters of the synthetic and the original time series data points in a 2D turbidity  $\times$  FDOM space generated by PCA (left) and t-SNE (right).

Training data	Test accuracy		
	R2	MAE	MSE
<b>Synthetic</b>	0.301858	0.016981	0.003859
<b>Real</b>	0.315577	0.016683	0.003672

Table 3: Test accuracy of RNN regression models trained on synthetic dataset and real dataset and then tested on real dataset.

with six units and a Sigmoid activation function. The optimizer used is Adam and the loss function is mean absolute error (MAE). Table 3 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 the machine learning models for the WTSD.

### 6.3. Anomaly detection performances of the best instances of the models

HF-PPAD generated best model instances and used synthetic datasets generated from the WTSD for training, and then tested the trained best model instances on the real dataset. Tables 4 and 5 summarize the performance results for each best trained model instance from the pool of models. All five models achieved high accuracy (70.2% to 97.3% for balanced accuracy and 64.7% to 94.6% for F-1 score across FDOM and turbidity), which

Model	Balanced accuracy	F-1 score	# of parameters	Training time	Epoch time	# of epochs
InceptionTime	97.3%	93.6%	1,817,888	350.5 sec	7 sec	50
ResNet	95.3%	90.1%	8,130,502	550.2 sec	11 sec	50
MiniRocket	93.4%	88.2%	89,974	150.6 sec	3 sec	50
LSTM	70.2%	64.7%	17886	50.8 sec	1 sec	50
TCN	90.7%	85.9%	68,556	60.2 sec	1.2 sec	50

Table 4: 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	# of parameters	Training time	Epoch time	# of epochs
InceptionTime	95.3%	89.9%	4,082,884	467.2 sec	9.34 sec	50
ResNet	98.3%	94.6%	11,921,636	721.5 sec	14.43 sec	50
MiniRocket	91.6%	85.1%	118,974	349.7 sec	6.94 sec	50
LSTM	74.2%	67.7%	23886	70.8 sec	1.4 sec	50
TCN	88.1%	81.9%	96,556	90.4 sec	1.8 sec	50

Table 5: Turbidity peak-pattern anomaly detection performance by the best trained model instance of each model in the HF-PPAD’s model pool.

indirectly affirms the best model generation ability of HF-PPAD. The computational costs varied more significantly than accuracy depending on the model. Notably, the best trained LSTM model instance, which achieved the lowest accuracy, also incurred the lowest computational cost. This observation confirms the trade-off that leads to user-provided performance preference addressed below in Section 6.4.

To further examine model performance with a focus on the anomaly detection accuracy, we have created the confusion matrices shown in Figure 8 for FDOM and Figure 9 for turbidity. Overall, the detection accuracy for all peak-pattern anomaly types are very high, which demonstrates the efficacy of the best model instance generation and training using the synthetic dataset. Particularly, the accuracy for the peak-pattern anomaly types FSK and FPT are 100% for all the best model instances; we believe this accuracy is attributed to a long sequence of their anomaly instances that differentiate them from the other types of peak pattern anomalies. The accuracy for NAP is relatively lower than other anomaly types, as apparently some of them are mistaken as PP, PLP, or SKP peaks. Note that NAP is not an anomalous peak type.

#### 6.4. 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

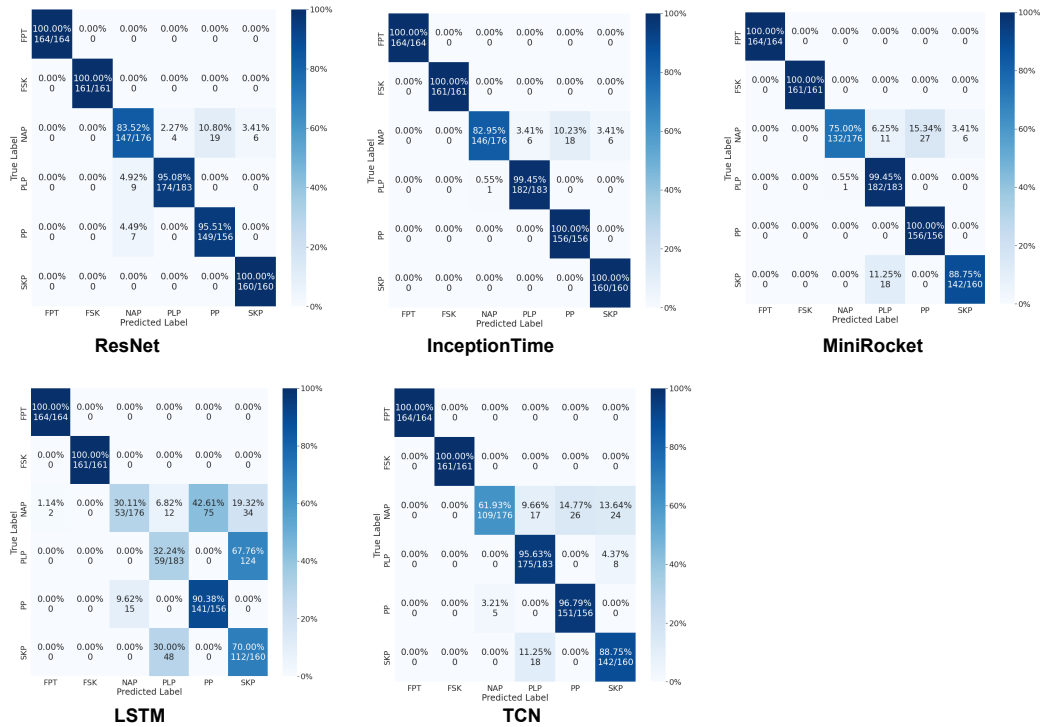


Figure 8: Confusion matrix of FDOM peak-pattern anomaly detection accuracy by the best trained model instance of each model in the HF-PPAD’s model pool.

Equation 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 Figure 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.

Figure 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 in-

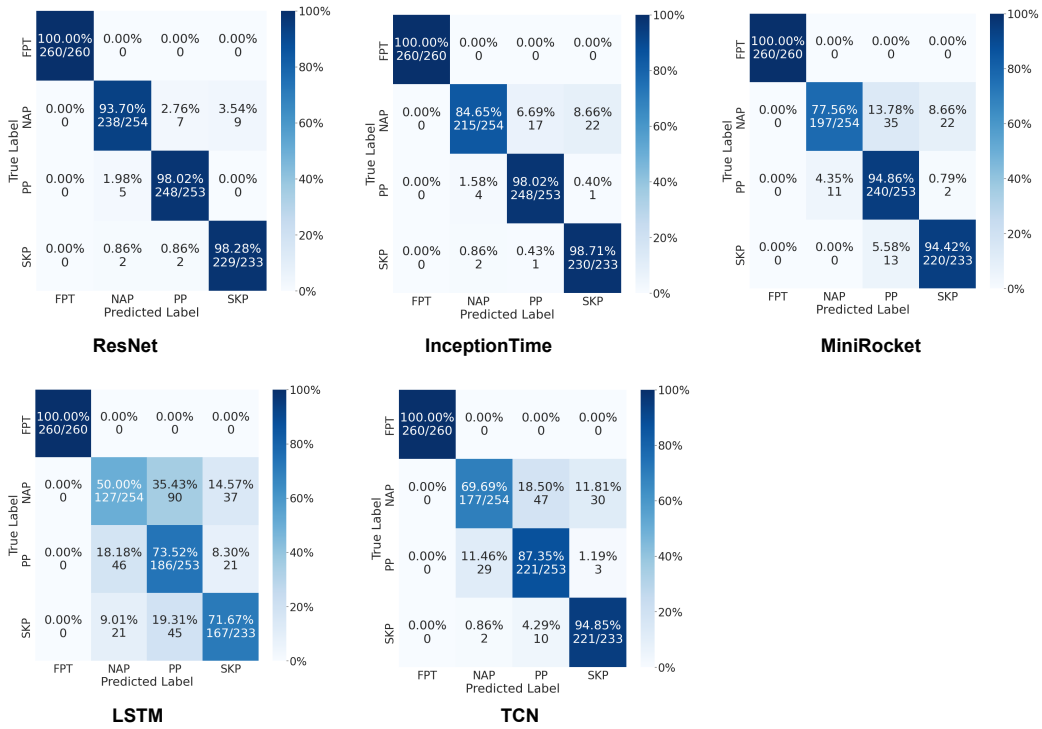
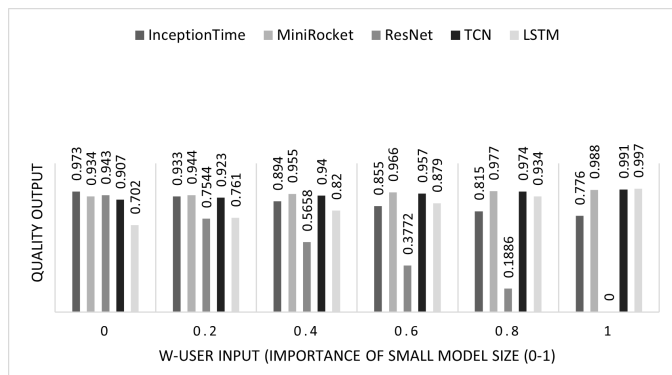


Figure 9: Confusion matrix of turbidity peak-pattern anomaly detection accuracy by the best trained model instance of each model in the HF-PPAD’s model pool.

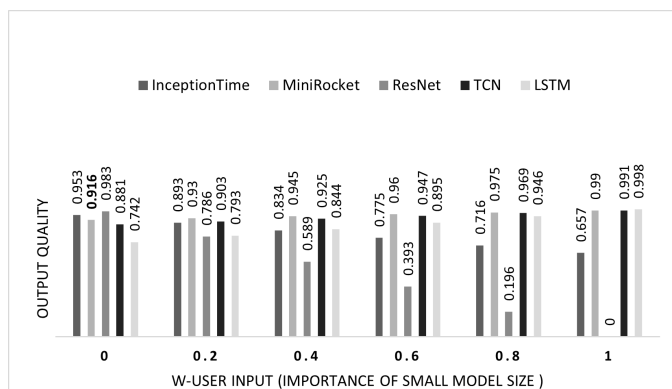
crement of 0.1). It visualizes the trends of the output qualities changing between the different models. Specifically, it exhibits a decrease in the output quality of a model with a larger size as the  $w$  value increases. Notably, the MiniRocket and TCN models are competitive options for users who 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. Conclusion

This paper presented an anomaly detection framework using automated machine learning (AutoML) on WTSD from the northeast US critical zone. The framework is designed to assist hydrologists in identifying anomalous



(a) For FDOM WTSD.

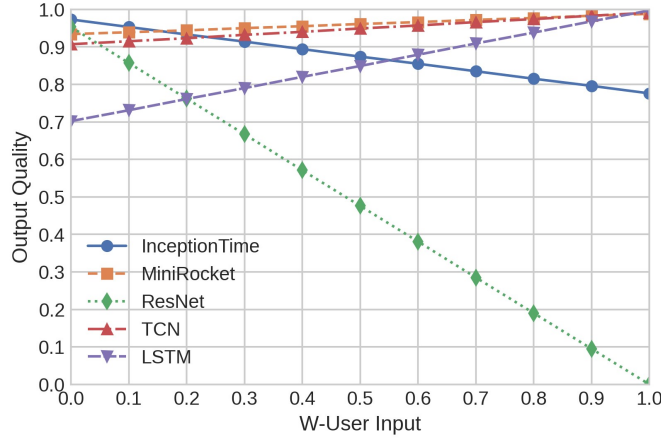


(b) For turbidity WTSD.

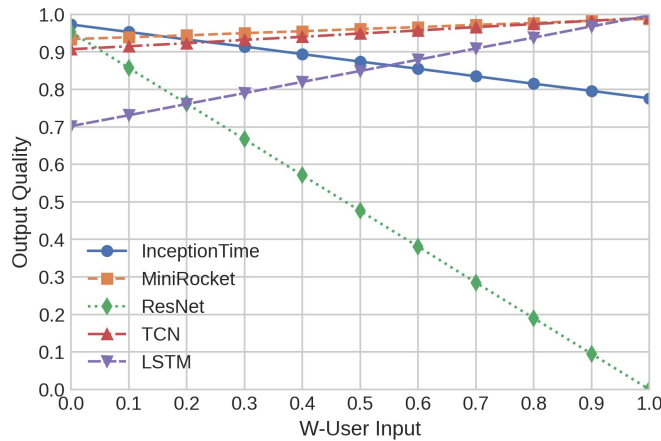
Figure 10: Comparison of the output quality achieved by the best model instance of each model for different values of the weight  $w \in [0, 1]$ ; the weight indicates how much the user prefers small model size to high accuracy.

events in their data, such as peak-pattern anomalies in FDOM and turbidity, without needing expert knowledge in machine learning or anomaly detection algorithms. The framework consists of two main components: a synthetic labeled dataset generator and an automated best model instance generator. During implementation, we used TimeGAN for the synthetic dataset generation, and used InceptionTime, ResNet, MiniRocket, TCN and LSTM as the models in the pool; then the model instance that is best overall considering both accuracy and computational cost (i.e. model size) was identified (for recommendation) according to the user preference.

Our work is the first to utilize automated machine learning for peak pat-



(a) For FDOM WTSD.



(b) For turbidity WTSD.

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

tern anomaly detection in WTSD. Our approach includes synthetically generated time series data and thorough hyperparameter optimization for model generation, demonstrating the potential of AutoML for time series classification tasks in hydrology. Experiments conducted demonstrate the high performance achieved by our framework applied to WTSD. Our contribution offers an innovative approach for efficient peak pattern anomaly detection in WTSD, providing a valuable tool for hydrologists and related stakeholders



in water management.

For future work, we plan to improve the framework by incorporating additional machine learning models and expanding the search space for model generation; we also plan to test the framework on a wider range of WTSD and other environmental sensors data (e.g., snow and air humidity) to validate its generalizability. Additionally, we plan to investigate the use of the framework for other domains of anomalous events, such as those observed in water quality monitoring and flood forecasting. By continuing to refine and expand the capabilities of the framework, we hope to make it an essential tool for hydrologists in their efforts to monitor and understand water resources.

## 8. 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

- Aggarwal, C. C. (2013). Outlier ensembles: position paper. ACM SIGKDD Explorations Newsletter, 14(2), 49-58.
- Bahri, M., Salutari, F., Putina, A., and Sozio, M. (2022). AutoML: state of the art with a focus on anomaly detection, challenges, and research directions. International Journal of Data Science and Analytics, 1-14.
- Chatterjee, S., Bopardikar, R., Guerard, M., Thakore, U., and Jiang, X. (2022). MOSPAT: AutoML based Model Selection and Parameter Tuning for Time Series Anomaly Detection. arXiv preprint arXiv:2205.11755.
- Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), 1-58.

- Cho, H., and 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.
- Dramsch, J. S. (2020). 70 years of machine learning in geoscience in review. *Advances in geophysics*, 61, 1-55.
- Dempster, A., Schmidt, D. F., & Webb, G. I. (2021, August). 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).
- Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and trends® in signal processing*, 7(3-4), 197-387.
- Evans, C., and 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.
- Enikeeva, F., and Harchaoui, Z. (2019). High-dimensional change-point detection under sparse alternatives. *The Annals of Statistics*, 47(4), 2051-2079.
- Feurer, M., Klein, A., Eggenberger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. *Advances in neural information processing systems*, 28.
- Fearnhead, P., and Rigall, G. (2019). Changepoint detection in the presence of outliers. *Journal of the American Statistical Association*, 114(525), 169-183.
- Fryzlewicz, P. (2014). Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6), 2243-2281.
- Feurer, M., and Hutter, F. (2019). Hyperparameter optimization. In *Automated machine learning* (pp. 3-33). Springer, Cham.
- Hamshaw, S., Denu, D., Holthuijzen, M., Wshah, S., & Rizzo, D. (2019). Automating the classification of hysteresis in event concentration-discharge relationships. In *Conference: SEDHYD 2019 conference, At Reno, Nevada*. [https://www.sedhy d. org/2019/openc onf/modul es/reque st. php](https://www.sedhyd.org/2019/openc onf/modul es/reque st. php).

- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- Haq, I. U., Khan, Z. Y., Ahmad, A., Hayat, B., Khan, A., Lee, Y. E., & Kim, K. I. (2021). Evaluating and Enhancing the Robustness of Sustainable Neural Relationship Classifiers Using Query-Efficient Black-Box Adversarial Attacks. *Sustainability*, 13(11), 5892.
- Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P. A. (2019). Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4), 917-963.
- Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., ... and Petitjean, F. (2020). Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6), 1936-1962.
- Jin, H., Song, Q., & Hu, X. (2019, July). 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).
- Jing, E., Zhang, H., Li, Z., Liu, Y., Ji, Z., and Ganchev, I. (2021). ECG heartbeat classification based on an improved ResNet-18 model. *Computational and Mathematical Methods in Medicine*, 2021.
- Khan, Z. Y., Niu, Z., Nyamawe, A. S., & ul Haq, I. (2021). A Deep Hybrid Model for Recommendation by jointly leveraging ratings, reviews and metadata information. *Engineering Applications of Artificial Intelligence*, 97, 104066.
- Kotthoff, L., Thornton, C., Hoos, H. H., Hutter, F., & Leyton-Brown, K. (2019). Auto-WEKA: Automatic model selection and hyperparameter optimization in WEKA. *Automated machine learning: methods, systems, challenges*, 81-95.
- Koren, O., Hallin, C. A., Koren, M., & Issa, A. A. (2022). AutoML classifier clustering procedure. *International Journal of Intelligent Systems*, 37(7), 4214-4232.

- Kotlar, M., Punt, M., Radivojević, Z., Cvetanović, M., & Milutinović, V. (2021). Novel meta-features for automated machine learning model selection in anomaly detection. *IEEE Access*, 9, 89675-89687.
- Kancharla, A., & Raghu Kishore, N. (2022). Applicability of AutoML to Modeling of Time-Series Data. In *Proceedings of Sixth International Congress on Information and Communication Technology* (pp. 937-947). Springer, Singapore.
- Lin, Y., Lee, B. S., and Lustgarten, D. (2019). Continuous detection of abnormal heartbeats from ECG using online outlier detection. In *Annual International Symposium on Information Management and Big Data* (pp. 349-366). Springer, Cham.
- Li, H., and Boulanger, P. (2020). A survey of heart anomaly detection using ambulatory Electrocardiogram (ECG). *Sensors*, 20(5), 1461.
- Lai, K. H., Zha, D., Wang, G., Xu, J., Zhao, Y., Kumar, D., ... and Hu, X. (2021, May). Tods: An automated time series outlier detection system. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 35, No. 18, pp. 16060-16062).
- Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1), 6765-6816.
- Li, Y., Zha, D., Venugopal, P., Zou, N., and Hu, X. (2020, April). Pyodds: An end-to-end outlier detection system with automated machine learning. In *Companion Proceedings of the Web Conference 2020* (pp. 153-157).
- Lai, K. H., Zha, D., Wang, G., Xu, J., Zhao, Y., Kumar, D., ... and Hu, X. (2021, May). Tods: An automated time series outlier detection system. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 35, No. 18, pp. 16060-16062).
- Li, P., Rao, X., Blase, J., Zhang, Y., Chu, X., & Zhang, C. (2021, April). CleanML: a study for evaluating the impact of data cleaning on ML classification tasks. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)* (pp. 13-24). IEEE.

- Meira, J., Eiras-Franco, C., Bolón-Canedo, V., Marreiros, G., & Alonso-Betanzos, A. (2022). Fast anomaly detection with locality-sensitive hashing and hyperparameter autotuning. *Information Sciences*, 607, 1245-1264.
- Matar, M., Xu, B., Elmoudi, R., Olatujoye, O., & Wshah, S. (2022). A Deep Learning-Based Framework for Parameters Calibration of Power Plant Models Using Event Playback Approach. *IEEE Access*, 10, 72132-72144.
- Neutatz, F., Chen, B., Alkhatib, Y., Ye, J., & Abedjan, Z. (2022). Data Cleaning and AutoML: Would an optimizer choose to clean?. *Datenbank-Spektrum*, 1-10.
- Pelletier, C., Webb, G. I., and Petitjean, F. (2019). Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing*, 11(5), 523.
- Pang, Guansong, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. "Deep learning for anomaly detection: A review." *ACM Computing Surveys (CSUR)* 54, no. 2 (2021): 1-38.
- Pellerin, B. A., Saraceno, J. F., Shanley, J. B., Sebestyen, S. D., Aiken, G. R., Wollheim, W. M., and Bergamaschi, B. A. (2012). Taking the pulse of snowmelt: in situ sensors reveal seasonal, event and diurnal patterns of nitrate and dissolved organic matter variability in an upland forest stream. *Biogeochemistry*, 108(1), 183-198.
- Qin, Y., and Lou, Y. (2019, March). Hydrological time series anomaly pattern detection based on isolation forest. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)* (pp. 1706-1710). IEEE.
- Ryzhikov, A., Borisyak, M., Ustyuzhanin, A., and Derkach, D. (2019). Normalizing flows for deep anomaly detection. *arXiv preprint arXiv:1912.09323*.
- Sun, J., Lou, Y., and Ye, F. (2017, November). Research on anomaly pattern detection in hydrological time series. In *2017 14th Web Information Systems and Applications Conference (WISA)* (pp. 38-43). IEEE.

- Schmidl, S., Wenig, P., and Papenbrock, T. (2022). Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9), 1779-1797.
- Sawaki, R., Sato, D., Nakayama, H., Nakagawa, Y., & Shimada, Y. (2019). ZF-AutoML: An Easy Machine-Learning-Based Method to Detect Anomalies in Fluorescent-Labelled Zebrafish. *Inventions*, 4(4), 72.
- Senagi, K. M. (2019). Random Forest Hyperparameter Optimization, GPU Parallelization and Applications to Soil Analysis for Optimal Crop Production (Doctoral dissertation, Paris 8).
- Singh, P., and Vanschoren, J. (2022). Meta-Learning for Unsupervised Outlier Detection with Optimal Transport. arXiv preprint arXiv:2211.00372.
- Shanley, J. B., Sebestyen, S. D., McDonnell, J. J., McGlynn, B. L., and 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.
- Shanley, J. B., Chalmers, A. T., Denner, J. C., Clark, S. F., Sebestyen, S. D., Matt, S., and Smith, T. E. (2022). Hydrology and biogeochemistry datasets from Sleepers River Research Watershed, Danville, Vermont, USA. *Hydrological Processes*, 36(2), e14495.
- Shende, M. K., Feijoo-Lorenzo, A. E., & Bokde, N. D. (2022). cleanTS: Automated (AutoML) Tool to Clean Univariate Time Series at Microscales. *Neurocomputing*.
- Tveten, M., Eckley, I. A., and 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.
- Vaughan, M. C., Bowden, W. B., Shanley, J. B., Vermilyea, A., Sleeper, R., Gold, A. J., ... & Schroth, A. W. (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.

- Wu, Y., Xi, X., & He, J. (2022). AFGSL: Automatic Feature Generation based on Graph Structure Learning. *Knowledge-Based Systems*, 238, 107835.
- Xing, H., Xiao, Z., Qu, R., Zhu, Z., & Zhao, B. (2022). An efficient federated distillation learning system for multitask time series classification. *IEEE Transactions on Instrumentation and Measurement*, 71, 1-12.
- Xiao, Z., Xu, X., Xing, H., Song, F., Wang, X., & Zhao, B. (2021). A federated learning system with enhanced feature extraction for human activity recognition. *Knowledge-Based Systems*, 229, 107338.
- Yu, Y., Wan, D., Zhao, Q., and Liu, H. (2020). Detecting pattern anomalies in hydrological time series with weighted probabilistic suffix trees. *Water*, 12(5), 1464.
- Yoon, J., Jarrett, D., and Van der Schaar, M. (2019). Time-series generative adversarial networks. *Advances in neural information processing systems*, 32.
- Yao, Q., Wang, M., Chen, Y., Dai, W., Li, Y. F., Tu, W. W., ... & Yu, Y. (2018). Taking human out of learning applications: A survey on automated machine learning. *arXiv preprint arXiv:1810.13306*.
- Zhao, Y., Nasrullah, Z., and Li, Z. (2019). Pyod: A python toolbox for scalable outlier detection. *arXiv preprint arXiv:1901.01588*.
- Zha, D., Lai, K. H., Wan, M., and Hu, X. (2020, November). Meta-AAD: Active anomaly detection with deep reinforcement learning. In *2020 IEEE International Conference on Data Mining (ICDM)* (pp. 771-780). IEEE.