Journal of Engineering Research

Volume 7 | Issue 1 Article 18

2023

T2D2: A Time Series Tester, Transformer, and Decomposer Framework for Outlier Detection

Abdel Hamid Khattab

Journal of Engineering Research (ERJ)
Vol. 7 – No. 1, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

T2D2: A Time Series Tester, Transformer, and Decomposer Framework for Outlier Detection

Abd Alhamid Khattab[1], Mahmoud Fahmy[2], and Nada Elshenawy[3]

[1]Computer and Automatic Control Dep., Faculty of Engineering, Tanta University, Tanta, Egypt - email: PG_38853@f-eng.tanta.edu.eg
[2]Computer and Automatic Control Dep., Faculty of Engineering, Tanta University, Tanta, Egypt - email: m.fahmy@f-eng.tanta.edu.eg
[3]Computer and Automatic Control Dep., Faculty of Engineering, Tanta University, Tanta, Egypt - email: nada_elshennawy@f-eng.tanta.edu.eg

Abstract: **Anomaly detection in time series has become an important aspect of data analysis and has many applications. Anomaly detection is often a challenge for statistical and pattern detection modeling. We introduce TTDD (Test, Transform, Decompose, and Detection), a pattern-based detector to detect outlier values in time series datasets. TTDD splits each time series into three components, each representing an underlying pattern category. Trend, seasonality, and residual. The outlier is determined for each component separately, which contributes to determining the outlier with high accuracy. The lag in the timing between the test set and the training set was discovered using Fourier transforms. In contrast to previous work, we support time series that are phase shifted. TTDD is resilient to consecutively outlier values with different types. The experimental results are based on real-world and benchmark datasets. The proposed TTDD framework outperforms the other methods with higher accuracy score at 95% which is higher than those of other methods about 15%. Furthermore, it can detect the delay time outliers that other methods cannot easily identify.**

Keywords: **Time series, Data quality, Anomaly detection, outlier detection**

I. INTRODUCTION

The flow of data became very large, that recent advances in technology have made it possible to collect a large amount of data over time which form time series. Data mining and decision-making systems relying on these time series need high-quality data to reduce the occurrence of errors and risks and improve results. One of the most important factors affecting the quality of data is the presence of anomalies. An anomaly is a data point that is significantly different from the remaining data. Anomalies are also referred to as abnormalities, deviants, or outliers in the data mining and statistics literature [1] Detecting abnormal values in time series data has become a crucial data mining task [2]. The detection of anomalies in the real-time flow of data has many applications such as credit card fraud detection, intrusion detection in cyber security, or preventive maintenance work, as well as medical applications and applications in information technology, energy, and social media [3], where the detection of anomalies can provide information on expected scenarios, limit the use of corrupted data for subsequent analysis, and allow operators to identify potential problems, and respond to it quickly and efficiently [4]. One of the major benefits of anomaly detection algorithms is their capability to detect unforeseen changes [4]. There are a number of outlier detection techniques introduced by the statistics community [3]. While statistical methods are mathematically more precise, they have several shortcomings, such as simplified assumptions about data representations, poor algorithmic scalability, and a low focus on interpretability [5]. Other anomaly detection algorithms find anomalies by understanding the distribution of their properties and isolating them from the rest of normal data samples [6]. The Isolation Forest is a promising and unique technique, data is sub-sampled, and processed in a tree structure based on random cuts in the values of randomly selected features in the dataset. Those samples that travel deeper into the tree branches are less likely to be anomalous [7]. Most of these techniques ignore to detect the structure of the dataset. The recent anomaly detection approaches can be relied upon for some kinds of time series but not for all series such as nonlinear and sinusoidal. Some time series consists of many components including trend, seasonality, and nonsystematic components, this makes it very difficult for the detection approaches to detect the outlier points. The paper aims to propose a framework that can deal with all kinds of time series having any components. In addition to its ability to detect outlier points that are difficult for the recent approaches to discover, such as the time delay and frequency shifts.

We argue that building an accurate performance model for outlier detection is not a trivial process. Hence, there is a need to critically examine the seasonality and the nature of the time series [8]. If the time series is a seasonal variation, the anomalous data point that is in between two “hills” may be categorized as a nominal point [6]. The current anomaly identification algorithms rely on studying a data profile to find samples that do not match that profile, it uses the fact that anomalous data are "few and different" [9], [10].

To address these challenges, in this paper, we develop TTDD (tester, transformer, decomposer, and detector). This framework can deal with any time series dataset without prior knowledge of its characteristics. It can test whether the time series is seasonal, trend, or stationary, In the case of the seasonal, trend results, the framework decomposes the data into its three main components: seasonal, trend and residual. Correctly decomposing the data needs to know the seasonal frequency in each unit of time, so we resort to the use of Fourier Transformer to extract the number of cycles in each time series, then the seasonal and trend compounds are set aside. The most professional algorithm (EIF) applied to the residual component, enhanced by adding a process, to detect the outlier threshold.

In addition to using the EIF, it was necessary to add an important part to the framework in order to be able to detect a certain type of outlier that the EIF is unable to identify, which is the different in frequency than the normal frequency as shown in the results in the case of UCR_Anomaly dataset, so

DOI: 10.21608/ERJENG.2023.190373.1148

Journal of Engineering Research (ERJ)
Vol. 7 – No. 1, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

there are two improvements over the EIF, namely, the data decomposition focus only on the component affecting the part about extracting the difference in frequency. Hence, this framework can extract the required characteristics and information from the dataset automatically and then efficiently detects the outlier according to this information. In the followings, we summarize the main contributions of the proposed work:

1) We introduce a five-component framework to automatically detect the outlier in time series datasets.
2) We propose a Test component to examine the dataset to define the classification of the dataset into one of three types, which are seasonal, trend, and stationary.
3) We employ a Transformer step making use of the Fourier Transform to get the frequency for each component in the time series [11].
4) Furthermore, we develop a Decomposer component to split the time series into: trend, seasonal, and residuals.
5) Detector: There are many algorithms for outlier detection,
6) Frequency comparator: it based on the Transformer results to get the frequencies of the training set and the test set and then compares between them and determines the periods of discrepancy in the frequency and considers them as outlier periods.
7) Frequency comparator: it based on the Transformer results to get the frequencies of the training set and the test set and then compares between them and determines the periods of discrepancy in the frequency and considers them as outlier periods.

The proposed framework is developed and evaluated to validate its performance using three datasets. The experimental evaluation demonstrates that the framework is capable of detecting the outlier professionally. The rest of the paper is organized as follows: the background and a set of related work is discussed in the following section. We then introduce an overview of the TTDD framework in Section 3. Section 4 illustrates the experimental evaluation and discusses the obtained results. Section 5 concludes the paper.

II. BACKGROUND

Detecting outliers is an active research topic. Outlier detection has been proven critical in many fields, such as credit card fraud analytics, network intrusion detection, and mechanical unit defect detection [1].. Therefore, many approaches have been proposed and several frameworks have been developed. We classify the related work into the following three categories.

A. *Tutorial and Survey*

The techniques on anomaly detection were grouped into different categories based on the underlying approach adopted by each technique [12].That happened after identifying the key assumptions, which were used by the techniques to differentiate between normal and anomalous behavior. For each category, the basic anomaly detection technique was provided, a discussion on the computational complexity of the techniques was provided too. An in-depth analysis of the current state of outlier analysis and data mining for IoT platforms in computer engineering involves examining recent developments in the field and exploring real-world applications that have been developed using data mining techniques. It also highlights the challenges associated with collecting, storing, and managing large datasets in the IoT environment, and discusses the various techniques that have been developed to overcome these challenges [13]. A meta-analysis of the anomaly detection problem has been provided, approaches to benchmarking anomaly detection algorithms that vary in their construction across several dimensions have been identified, the effects of experimental design on experimental results have been observed and results are evaluated. This analysis provides an ontology for describing anomaly detection contexts; a methodology for controlling various aspects of benchmark creation and guidelines for future experimental design [14].

A comprehensive and organized review of the progress of outlier detection methods presented [15]. In this survey, the fundamental concepts of outlier detection are offered and then the different techniques from diverse outlier detection techniques are categorized, such as distance-, clustering-, density-, ensemble-, and learning-based methods.

Unsupervised machine learning algorithms in the context of outlier detection have been studied through a variety of research fields to applications including fraud detection, intrusion detection, medical diagnoses, and data cleaning [16]. The selected methods have been benchmarked on publicly available datasets and novel industrial datasets. A full overview of the algorithms' characteristics has been built by submitting each method to extensive scalability, memory consumption, and robustness tests. A taxonomy for existing deep learning techniques based on their underlying assumptions was discussed [17]. The goal in this survey is to provide an easier better understanding of the techniques belonging to different categories in which research has been done on this topic and provide the relative strengths and weaknesses of the approaches. A comprehensive taxonomy covers advancements in three high-level categories and fine-grained categories of the methods [18]. It reviews their key intuitions, objective functions, underlying assumptions, advantages and disadvantages, and discusses how they address the aforementioned challenges.

Kai Ming Ting & Sunil introduced comparative works in anomaly detection and their recommendations; an analysis of strengths and weaknesses of current comparative works; current bias-variance analyses applied to anomaly detection; dealing with high-dimensional datasets and subspace anomaly detection and Factors to consider in choosing an anomaly detector [19]. A survey of contemporary techniques for outlier detection, the respective motivation for each technique, and advantages and disadvantages in a comparative review have been introduced [20].

B. *Key Algorithms*

We classified the outlier detection algorithms into four groups as shown in table 1. We briefly describe each group below.

Journal of Engineering Research (ERJ)
Vol. 7 – No. 1, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

B.1 Probabilistic algorithms

ECOD (Empirical-Cumulative-distribution-based Outlier Detection) [21] estimates the underlying distribution of the input data in a nonparametric fashion by computing the empirical cumulative distribution per dimension of the data, using these empirical distributions to estimate tail probabilities per dimension for each data point. Then ECOD computes the outlier score of each data point by aggregating estimated tail probabilities across dimensions. FastABOD [22] introduced an approach to outlier detection based on the variance of angles between pairs of data points. FastABOD is suitable for low-dimensional with big data sets and filtered the refinement for high-dimensional data. COPOD is an outlier detection algorithm, which constructs an empirical copula, and then uses it to predict tail probabilities of each given data point Linear Model to determine its level of “extremeness”, this work, proposes a parameter-free outlier detection algorithm [23] .

B.2 Linear Model

PCA [[24], [7]] proposes a scheme that uses a robust principal component classifier in intrusion detection problems where the training data may be unsupervised. Assuming that anomalies can be treated as outliers, an intrusion predictive model is constructed from the major and minor principal components of the normal instances. A measure of the difference of an anomaly from the normal instance is the distance in the principal component space. KPCA [44] is a non-linear extension of PCA. OCSVM [25] tries to estimate a function that is positive on the subset and negative on the complement. The function is given by a kernel expansion in terms of a potentially small subset of the training data; it is regularized by controlling the length of the weight vector in an associated feature space.

B.3 Proximity-Based

ROD [11] decomposes the full attributes space into different combinations of subspaces, in which the 3D vectors, representing the data points per 3D subspace, are rotated about the geometric median, using the Rodrigues rotation formula, to construct the overall outlying score. SOD [10] proposes an outlier detection schema that detects outliers in varying subspaces of high-dimensional feature space. In particular, for each object in the data set, we explore the axis-parallel subspace spanned by its neighbors and determine how much the object deviates from the neighbors in this subspace. HBOS [9] a histogram-based outlier detection (HBOS) algorithm scores records in linear time. Models univariate feature densities using histograms with a fixed or a dynamic bin width. Afterward, all histograms are used to compute an anomaly score for each data instance. A method that explicitly isolates anomalies instead of profiling normal points has been proposed [[26], [7]]. The use of isolation enables the proposed method, iForest, to exploit subsampling to an extent that cannot be feasible in epoch methods, creating an algorithm that has a linear time complexity with a low constant and a low memory requirement. iForest worked well in high-dimensional problems which have a large number of irrelevant attributes, and in situations where the training set does not contain any anomalies. In the case of the data having an inherent structure the iForest suﬀ ered from artifacts generated by the criteria for the branching operation of the binary tree. The Extended Isolated Forest (EIF) improved the iForest algorithm by transforming the data randomly before the creation of each tree, which results in averaging out the bias, allowing the slicing of the data to use hyperplanes with random slopes [27].

B.4 Neural Networks

The Autoencoder ensembles for unsupervised outlier detection were introduced [20]. The proposed approach showed that neural networks can be a competitive technique to other epoch methods after overcoming the accompanying problems such as the sensitivity to noise and required large data sets, and that are done by randomly varying the connectivity architecture of the Autoencoder to obtain significantly better performance. This technique is combined with an adaptive sampling method to make the approach more efficient and effective. DeepSVDD [28] introduces a Deep Support Vector Data Description, and jointly trains a deep neural network while optimizing a data-enclosing hypersphere in output space through this Deep SVDD extracts common temporal correlation of time series distributions. Instead of treating each data stream independently. Neural network, with their different types and developments, has been used in a wide variety of research into anomaly detection [29] [30] [31] [32]].

Table 1. Outlier detection algorithms

	Abbreviation	Algorithm
Probabilistic	ECOD [21]	Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions
	FastABOD [22]	Fast Angle-Based Outlier Detection using approximation
	COPOD [23]	COPOD: Copula-Based Outlier Detection
Linear Model	PCA [24]	Principal Component Analysis (the sum of weighted projected distances to the eigenvector hyperplanes)
	KPCA [27]	Kernel Principal Component Analysis
	Max–Min [7]	Max–Min Robust Principal Component Analysis
	OCSVM [25]	One-Class Support Vector Machines
Proximity-Based	HBOS [9]	Histogram-based Outlier Score
	SOD [10]	Subspace Outlier Detection
	ROD [11]	Rotation-based Outlier Detection
	IForest [12]	Isolation Forest
	EIF [6]	Extended Isolation Forest
	BS-iForest [26]	box plot-sampled iForest for wireless sensor networks
Neural Networks	Beta-VAE [29]	Variational AutoEncoder (all customized loss term by varying gamma and capacity)
	SO_GAAL [30]	Single-Objective Generative Adversarial Active Learning
	DeepSVDD [28]	Deep One-Class Classification
	MAN_GAN [38]	Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks
	ALAD [39]	Adversarially learned anomaly detection
	AutoEncoder Ensemble [40]	Outlier detection with autoencoder ensembles
	DNN [31]	A deep neural network (DNN) to handle the outlier detection problem in the context of streaming data
	BNNs [32]	Bayesian Neural Networks in Outlier Detection

DOI: 10.21608/ERJENG.2023.190373.1148

Journal of Engineering Research (ERJ)
Vol. 7 – No. 1, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

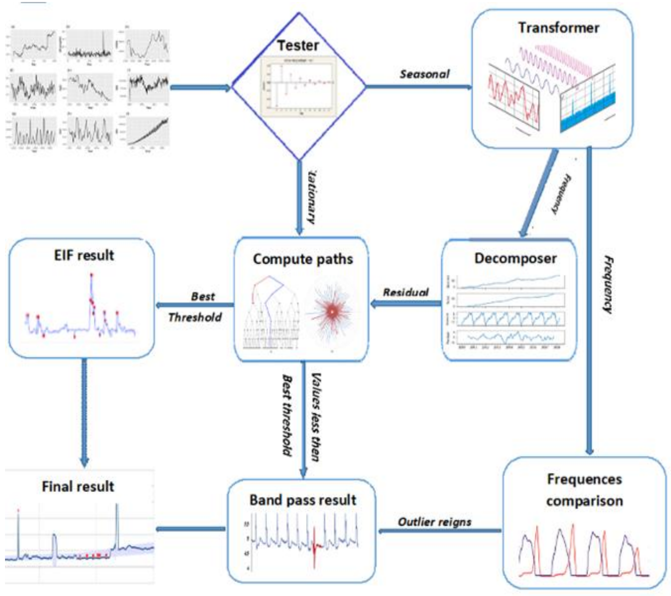


Figure. 1. Block diagram for TTDD (Tester, Transform, Decomposer, and Detector).

C. Applications

There are many application for the anomaly detection systems. One of these applications is improved based on Long Short-Term Memory (LSTM) networks to reduce operational risk in spacecraft monitoring systems [33]. A combination between SR and CNN is introduced to improve the performance of SR model [34] to tackle the problem of time-series anomaly detection. Social media is one of the most important anomaly detection applications. A survey of current approaches to addressing this problem has been proposed, focusing on the new type of anomaly in social media and reviewing newly developed techniques for detecting those special types of anomalies, while providing an overview of the problem area, common formulas, current methodologies and potential trends [35]. Qi Yu and Xinran proposed a hierarchical Bayes model: GroupLatent Anomaly Detection (GLAD) model [36]. GLAD takes both pairwise and point-wise data as input, automatically infers the groups, and detects group anomalies simultaneously. GLAD studies the collective behavior of individuals and detects group anomalies. For handling traffic flow distributions rather than individual observations an established outlier detection method has been adapted, the local outlier factor (LOF), this method applied the outlier detection online to extend the database with new flow distributions that are considered inliers [37].

III. THE PROPOSED FRAMEWORK: AN OVERVIEW

To cope with the previously mentioned challenges, in this section, we introduce a new framework, namely TTDD (Tester, Transform, Decomposer, and Detector). The TTDD is a five-phase framework, as depicted in fig. 1. In the first phase, Tester is used to test whether a given time series is stationary or not. When the time series is stationary it transfers directly to the fourth phase. It is not treated in the second and third stages. Some time series are non-stationary, it goes through the second phase, the Transformer, which provides frequency information about the series. The third stage, Decomposer, used the maximum frequency resulting from the Transformer phase as a *freq* parameter. The *freq* reciprocal is used in the decomposition function. This function separates the time series into three components: trend, seasonal, and residuals. The Residual component treaded in the fourth phase, Decoder, this component aims to detect outliers from the residual data, this is the main goal of the research, and all previous operations are data preparation. The detector has two processes: Compute paths and Best threshold detection. Through experiments, it has been shown that the EIF used in the fourth phase cannot deal well with the delay in timing. Therefore, it was necessary to add another stage to discover the delay in timing. This stage is the frequency comparison stage. In the following.

D. Tester

This component aims to test whether a given Time series is stationary or not by applying two statistical tests Augmented Dickey Fuller (ADF) and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) [41]. A stationary time series is one whose statistical properties such as mean, variance, and auto- covariance are all constant and not a function of time [42]. Checking the stationarity of the series can be done in several methods. The simplest of these is just looking at the plots to review the time series diagram of the data and visually check if there are any clear trends or seasonality. The second method is the Autocorrelation function (ACF), the autocorrelation is intended to measure

The relationship between a variable's present value and any past values that were accessed before. The ACF method defined the sample auto-covariance function (ACVF) as C_k in eq1, where x_t is the current observation, $x_{(t+k)}$ is the lagged observation at prior time steps k, and x is the mean value [41].

$$Ck = \left(\frac{1}{n}\right)\sum_{t=1}^{n-k}(x_t-\overline{x})\,(x_{t+k}-\overline{x}) \tag{1}$$

Then the sample autocorrelation function (ACF) is defined as eq. (2):

$$rk = \left(\frac{Ck}{C0}\right) = Cor(x_t, x_{t+k}) \tag{2}$$

Where C_0 is the auto-covariance of $\{x_t\}$ at lag 0. Figure 2 shows an example for the ACF plot.

There are several tests to check stationarity such as Dickey-Fuller (DF), ADF, Phillips-Perron (PP), and (KPSS) [43] .The Dickey-Fuller test is Test for "unit root" as shown in eq. (3)

$$y't = \varphi yt - 1 + b1y't - 1 + b2y't - 2 \cdot\cdot + bky't - k \tag{3}$$

Where y′t denotes differenced series yt−yt−1. yt is already stationary if ˆφ <0. The Number of lagged terms, k, is usually set to be about three.

Similar to the original Dickey-Fuller, the augmented Dickey-Fuller tests for a unit root in a time series sample. The primary differentiator between the two tests is that the ADF is utilized for a larger and more complicated set of time series

Journal of Engineering Research (ERJ)
Vol. 7 – No. 1, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

models [41]. The augmented Dickey-Fuller statistic used in the ADF test is a negative number. The more negative it is, the stronger the rejection of the hypothesis that there is a unit root. The Augmented Dickey-Fuller test allows for higher-order autoregressive processes by including Δyt−p in the model. As in the following model eq. (4):

$$yt \;= \alpha + \beta t + \;\; yt - 1 + \delta 1 \Delta yt - 1 + .. + \delta p \Delta yt - p \qquad (4)$$

Where α is a constant, β is the coefficient on a time trend and p is the lag order of the process.

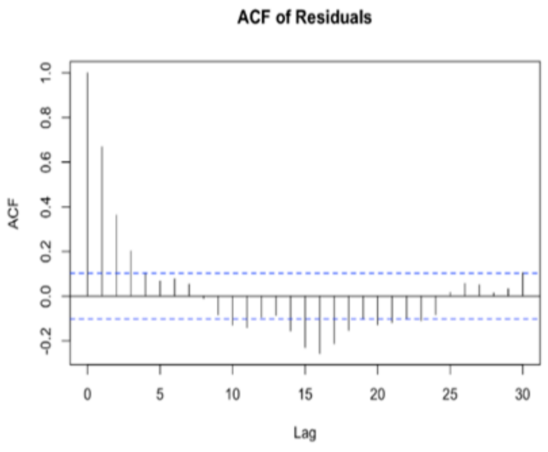


Figure 2. ACF plot.

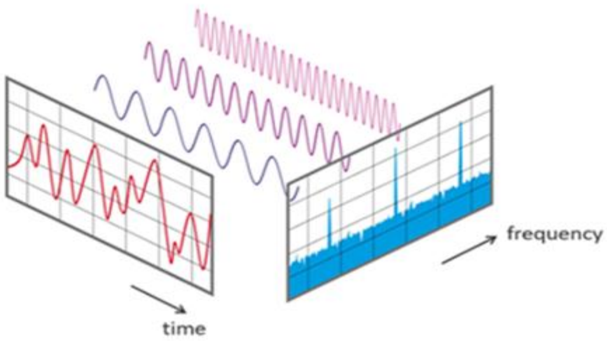


Figure. 3. Series in the time and frequency domain.

Table 2. Tester output

ADF	KPSS	Tester
not stationary	not stationary	not stationary
stationary	stationary	stationary
stationary	non-stationarity	difference stationary
non-stationarity	stationarity	trend stationary

By including lags of the order p the ADF formulation allows for higher-order autoregressive processes. This means that the lag length p has to be determined when applying the test. One possible approach is to test down from high orders and examine the t-values on coefficients. The intuition behind the test is that if the series is characterized by a unit root process then the lagged level of the series (yt-1) will provide no relevant information in predicting the change in yt besides the one obtained in the lagged changes Δyt-1. In this case the γ=0 and null hypothesis is not rejected. In contrast, when the process has no unit root, it is stationary and hence exhibits reversion to the mean, so the lagged level will provide relevant information in predicting the change of the series and the null of a unit root will be rejected. The null and alternate hypothesis for the KPSS test are opposite that of the ADF test, Null Hypothesis means the process is trend stationary where the alternate hypothesis means the series has a unit root (series is not stationary).

We apply the ADF and the KPSS tests to ensure that the series is truly stationary. The Tester outcome as shown in table 2, in cases three and four due to the difference in the results from the ADF test and the KPSS test. The series detrended by differencing [44]. Differencing means a new series is constructed where the value at the current time step is calculated as the difference between the original observation and the observation at the previous time step.

E. Transformer

After the tester determines that the series is non-stationarity the series is forwards to the Transformer. We apply the (FFT) as fast algorithm, low complexity, for the computation of the discrete Fourier transform (DFT), it is considered as a special case of the Fourier transform [45], FFT reduces the complexity of computing the DFT from O (N^2) to O (N log N), where N is the data size. So we apply the Fast Fourier Transformation to convert the series into individual spectral components and thereby provides frequency information about the series as shown in fig. 3.

Fourier analysis is a method for expressing a function as a sum of periodic components, FT of x(n) is defined as eq. (5):

$$X\,(k) \;= \sum_{k=0}^{N-1} x(n) e^{-j2\pi kn/N} \quad \leq k \geq N-1 \qquad (5)$$

Our Transformer model decomposes the series into components of different frequencies as shown in eq 5, then it selects the maximum component in the amplitude to transfer its frequency to the Composer as a *freq* parameter.

F. Decomposer

This component aims to divide the data series into three components: the estimated trend component, the estimated seasonal component, and the estimated residuals. Decomposition provides a better understanding for the time series analysis and forecasting, The Decomposer provides two decomposition methods: Additive Decomposition and Multiplicative Decomposition. In the Additive one, the relationship is formed as *Xt*= *Tt*+ *St*+ *Rt*, while in the Multiplicative one the relationship is formed as *Xt*= *Tt* . *St*. *Rt*. The Decomposer results are obtained by first estimating the trend by applying a convolution filter to the data. The trend is then removed from the series and the average of this de-trended series for each period is the returned seasonal component [46]. The moving average method used in filtering can be written as eq. (6):

$$Tt \;= \frac{1}{m} \sum_{j=-k}^{k} yt + j \qquad (6)$$

where m=2k+1. That is, the estimate of the trend-cycle at time

DOI: 10.21608/ERJENG.2023.190373.1148

Journal of Engineering Research (ERJ)
Vol. 7 – No. 1, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

t is obtained by averaging values of the time series within k periods of t .

Each component was separately plotted and saved as shown in fig. 4. The plot gives us an overview of the Behavior and properties of the series, our interest in the following stages will be the Residual component, as the other two components represent the physical properties of the series and is not contain any information about the outlier values.

G. *Detector*

This component aims to detect outliers from the residual data, this is the main goal of the research, and all previous operations are data preparation. Handling residual data enable us to deal with any types of series whether this series trend, seasonal, stationary, any complex cases such as anomalous data between two "hills". Detector contains two processes: compute paths and best threshold detection.

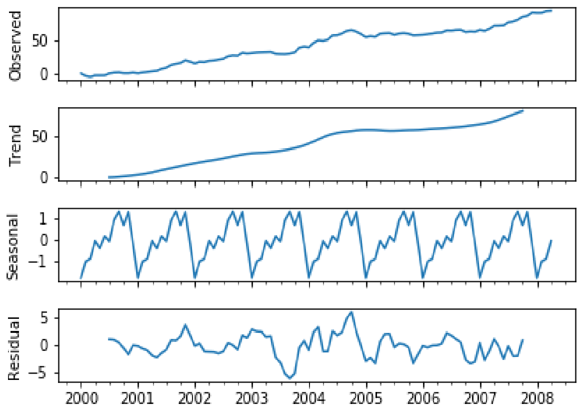


Figure. 4. Decomposition the series to its components trend, seasonal, stationary.

1) Compute paths

In this process we compute anomaly scores for all data points depending on the Extended Isolation Forest Compute path function [6]. In the beginning of the EIF training process, a recursively partitioning to the data series by randomly selecting any attribute and value for that attribute within the minimum and maximum values in that dimension, till the singleton node is reached (leaf). This partitioning is repeated multiple times in order to create a forest as shown in figure 5. The red points depict anomalous points traveling down the tree, while the blue line shows that of a nominal point. The anomalous point is isolated very quickly, but the nominal point travels all the way to the maximum depth. The depth each point reaches in each trained tree is computed and converted to an anomaly score. Outliers are marked with low anomaly scores where normal points are marked with high ones, this requires setting a good threshold to filter out anomalies. That was our motivation to develop the next process- Best Threshold Detection.

2) Best Threshold Detection

Precision-Recall is a useful measure of the success of prediction, in this method we compute the precision-recall pairs for different probability thresholds. The precision is the ratio:

$$\text{Precision} = \frac{Tp}{(Tp+Fp)} \qquad (7)$$

where *tp* is the number of true positives and *fp* the number of false positives.

Precision is a measure of result relevance. The recall is the ratio:

$$\text{Precision} = \frac{Tp}{(Tp+Fn)} \qquad (8)$$

which measures how many truly relevant results are returned [47].

After computing the precision and the recall a tradeoff is made between accuracy and recall for different thresholds. The high precision relates to the low false positive rate, and the high recall relates to the low false negative rate. High scores for both show that the classifier returns accurate results (high resolution), as well as returning the majority of positive results (high recall). The score is interpreted as a weighted average of the precision and recall, where the score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the score are equal. The formula for the score is:

$$F1 = \frac{2*(precision*recall)}{(precision+recall)} \qquad (9)$$

After all possible thresholds have been tried, the highest score is the opposite of the best threshold. Then we use this threshold to separate outliers from normal values.

3) Frequency comparison

Through experiments, it has been shown that the EIF cannot deal well with the delay in timing. Therefore, it was necessary to add another stage to discover the delay in timing [43] after discovering the outlier values using the EIF in the previous stages. The Transformer is used again to determine the frequencies of both the test and the training set, that's after separating the outlier values resulting from the EIF, then we compare the frequencies over the period and identify the periods that are most different, and determine the minimum and maximum frequency in those periods. Depending on the Band-pass filtering, the outlier values are determined in the time domain [48].

IV. EXPERIMENTAL EVALUATION

To demonstrate the effectiveness of the TTDD framework, we conducted a set of experiments comparing our solution to recently developed approaches. In the following, the experimental setup, experimental methodologies, and experimental results will be presented.

A. *Experimental setup*

We ran our experiments in a server with Core i7 Intel processors 2.60 GHz, 8 GB RAM and 250 GB SATA hard drive running Windows 10. The proposed framework was implemented in the open source Python package TTDD, which provides a list of transformation and outlier scoring methods with visualization and performance evaluation techniques. The package was used is presented and available from Github

DOI: 10.21608/ERJENG.2023.190373.1148

Journal of Engineering Research (ERJ)
Vol. 7 – No. 1, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

(github.com/Eng-khattab/TTDD). The three datasets used for this article are also available.

B. Experimental methodology

1) Selecting the datasets

We used three time series datasets as mentioned. These Datasets were selected in a way that illustrates the abovementioned ideas in this research, where they differ in size and whether stationary or not as shown in table 3. It will be abbreviated as follows:

The first dataset is one of the UCR Time Series Anomaly Archive, The benchmark which deals with the four defects common in other benchmarks. Where the popular benchmark datasets suffer from one or More than four defects. These flaws are trivial and unrealistic Anomaly intensity, misleading ground truth and running to failure bias, tendency [40]. We used The UCR_Anomaly_BIDMC1_2500_5400_5600, a dataset from the UCR archive. The first 2,500 data points (the '2500' in the file's name) are designed to be used as training data, and the anomaly itself is located between data-points 5,400 and 5,600 (the '5400_5600' in the file's name) indicate the location of the anomaly. While evaluating this dataset by KPSS and Augmented Dickey-Fuller Testers, the result.

The second dataset is the Oddwater dataset; a water-quality dataset collected using in situ sensors a natural river system [3]. Three series which are turbidity, conductivity and river level, obtained from Pioneer River from 12 March 2017 to 12 March 2018, these datasets include 6280 recorded points. The water-quality experts labeled the observations as either outliers or not. While evaluating this dataset by Augmented Dickey-Fuller Tester, the result showed that the dataset is stationary, but while evaluating it by KPSS_test tool, it was not stationary. So we considered this dataset as not stationary as shown table 3.

The third dataset is the Safe Drinking Water Act data (SDWA) Dataset. The United States Environmental Protection Agency (EPA) designed a website to provide easy Access to EPA's compliance and enforcement data [49] . Large datasets can be downloaded and used for many different functions and 1certain to meet all data retrieval needs. The SDWIS dataset was chosen, as it was the most commonly used. In the enforcement and compliance program. By using the ADF and KPSS to evaluate the dataset behavior, the two tools gave a stationary result.

2) Extract the main frequency components

The Fourier Transformer is used to extract the main frequency for the non-stationary dataset. This stage applied only on the non-stationary dataset after testing it by ADF and KPSS testers. The stationary dataset such as SDWA has zero frequency, so we do not apply this stage on the stationary datasets. Alternatively, we apply it on the non-stationary dataset in order to get the suitable frequency for each dataset which will be used in the next stage.

3) Split each time series to its basic components

Decomposing the time series is an important step in improving the modeling accuracy. After we got the main frequency to the non-stationary dataset in the previous stage we used it in the decomposition function. Decomposition splits the time series data into three components: trend, seasonal and residual. Trend and seasonal represent the normal characteristics of the dataset, so we concentrate our attention on as shown if fig. 6.

4) Calculate outlier scores

The Extended Isolated Forest is used to compute the anomaly score for each point [6], the data is sub-sampled, and processed in a tree structure based on random cuts in the values of randomly selected features in the dataset. Those samples which travel deeper into the tree branches are less likely to be anomalous, while the aggregated lengths of the tree branches is used as a metrics for measuring the anomaly score for every given point

5) Determine the optimal threshold

ROC Curves and Precision-Recall Curves are used to select the best or optimal threshold directly [50]. A grid search is used to tune the threshold and locate the optimal value; the value where there is high true positive rate and low false positive rate. The optimal threshold function in the TTDD frame scores the points as an anomaly or outlier depending on the optimal threshold.

6) Determine the delay in timing:

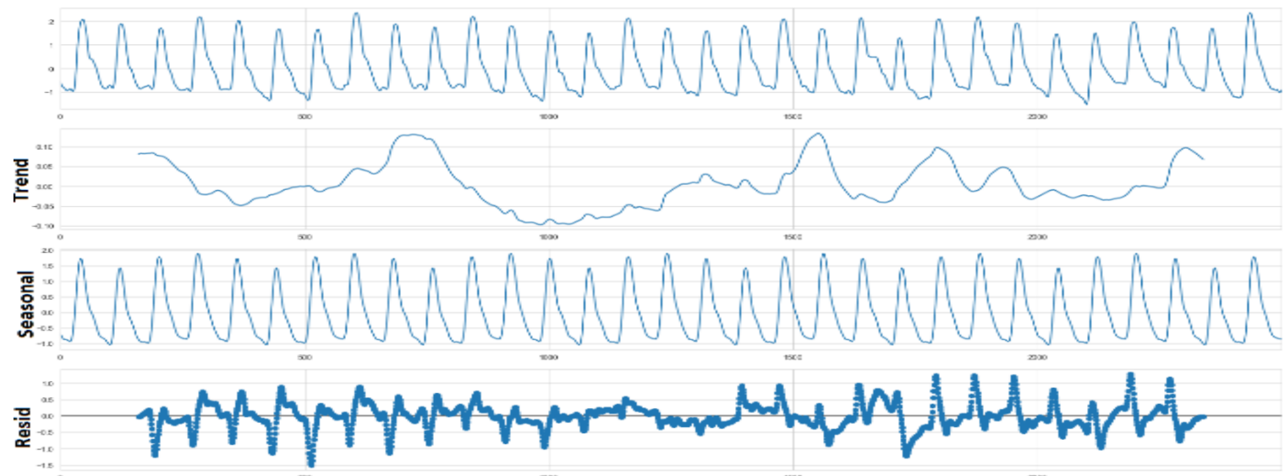
The Fast Fourier Transform (FFT) algorithm samples the training set and the test set over the period of the test space and divides it into its frequency components. The Frequency components of the test set are compared with the frequency components of the training set, and the periods of significant difference in the amplitude are determined. Then the Inverse Fast Fourier Transform (IFFT) is used to go back to the time domain. The aforementioned periods are highlighted with different color, they are considered as outlier periods. Fig7 shows the outputs for this step.

Table 2. Datasets features

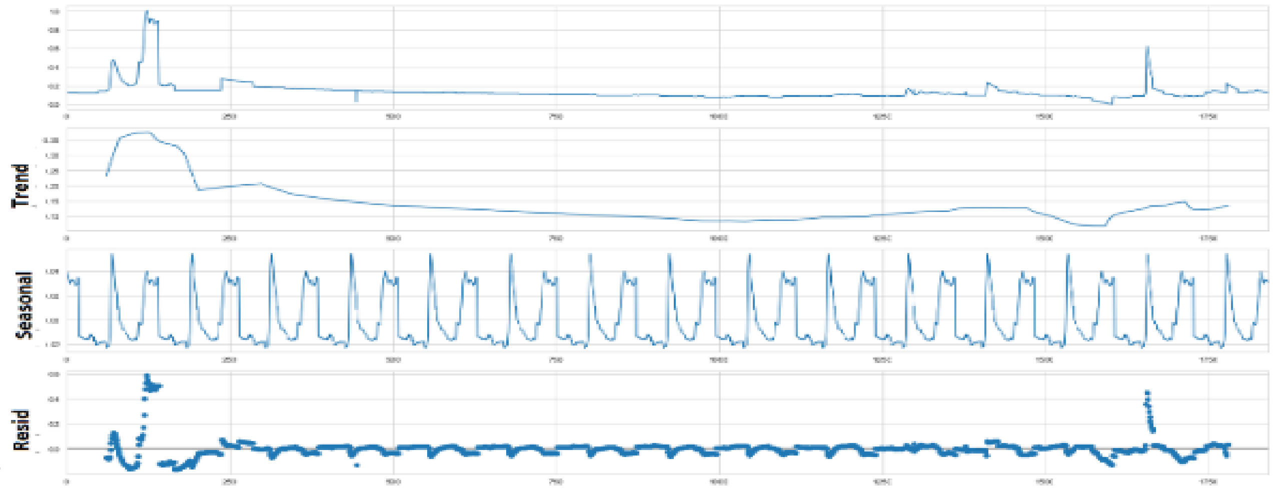
Dataset	Rows	Columns	ADF result	KPSS result	TTDD result
UCR_Anomaly	6000	1	Not stationary	Not stationary	Not stationary
Oddwater	6280	7	stationary	Not stationary	Not stationary
SDWA	1048576	2	stationary	stationary	stationary

DOI: 10.21608/ERJENG.2023.190373.1148

Journal of Engineering Research (ERJ)
Vol. 7 – No. 1, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873



6.a. UCR decomposition to its main components



6.b. Oddwater (level) components

Figure 6 Decomposition Stage Output

7) *Model accuracy calculation*

We measure the accuracy of the TTDD framework using 3 types of error measurements:

- The ROC AUC: Area under a Receiver Operating Characteristics (ROC) Curve (AUC), the ROC curve is a two-dimensional depiction of classifier performance [51] . Since the AUC is a portion of the area of the unit square, its value will always be between 0 and 1.0. Using the opposite matrix, the AUC formula extracted through the following steps: In the beginning the sensitivity and the specificity were calculated:

$$Sensitivity = \frac{Tp}{(Tp+Fn)} \qquad (10)$$

$$Specificity = \frac{Tn}{(Tn+Fn)} \qquad (11)$$

The sensitivity indicates what proportion of the positive class got correctly classified, the specificity indicates what proportion of the negative class got correctly classified.

$$AUC = \frac{Tp}{(sensitivity+specificity)/2} \qquad (12)$$

- F1_score: the F1 score conveys the balance between the sensitivity and the specificity [52]. Its formula is:

$$F1\_score = 2 * \left(\frac{sensitivity*specificity}{sensitivity*specificity}\right) \qquad (13)$$

- Accuracy score: this score computed by the formula:

$$\text{Accuracy} = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(y^{\wedge} = yi) \qquad (14)$$

where y^ is the predicted value and y is actual value.

C. *Experimental results and Comparison*

The TTDD framework has been employed to detect the outlier. We utilize three time series this research shown in table 3. While the UCR_Anomaly and Oddwater are selected as non-stationary datasets, the SDWA is selected as a clear stationary dataset.

D. *TTDD performance*

The TTDD framework can deal efficiently with seasonal data, as shown in figure 7. The UCR_Anomaly dataset, a clear seasonal dataset, confirmed by the results of the two stationary tester (KpSS, ADF), has a slight lag in the timing as show in the yellow rectangle in figure 7.a. the isolated forest IF and the extended isolated forest cannot detect this lag as shown in

Journal of Engineering Research (ERJ)
Vol. 7 – No. 1, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

figure 7.b and datasets as mentioned. These Datasets were selected in a way that illustrates the abovementioned ideas in 7.c, but the TTDD can detect this lag as shown in the yellow rectangle in figure 7.d. We have applied the TTDD model to the UCR_Anomaly with different training parameters and different test sets. Table 4 shows the output for each function in the TTDD for this dataset.

We applied another database experiment which is the Oddwater database, the results show that the TTDD can detect the outliers as shown in figure 8.d in a way that is very close to the truth which represented in figure 8.a. Table 4 shows the output for each function in the TTDD for this dataset. Finally, we applied the TTDD to the SDWA dataset, which is considered a special case as it is stationary. Therefore, TTDD does not call the decomposition and Fourier Transformation functions and is considered as if it was an EIF model AS shown in figure 9.

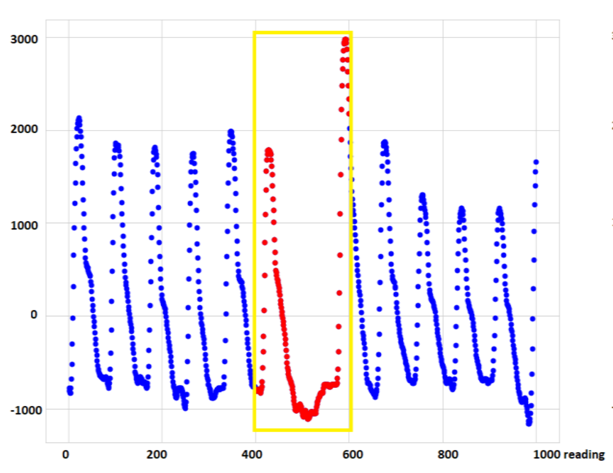


Figure 7 .a UCR_Anomaly test set

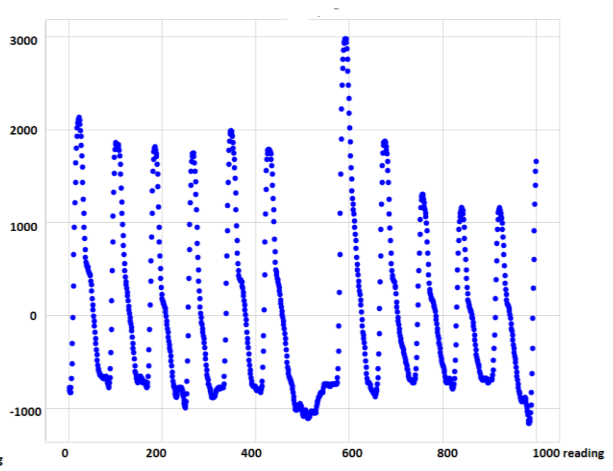


Figure 7.b IF output

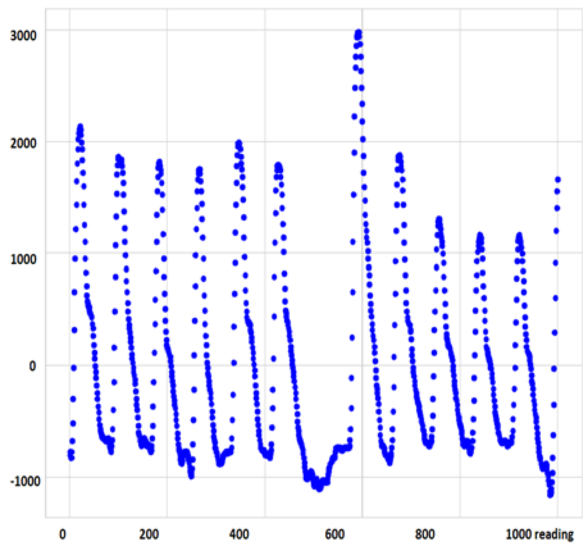


Figure 7 .c EIF output

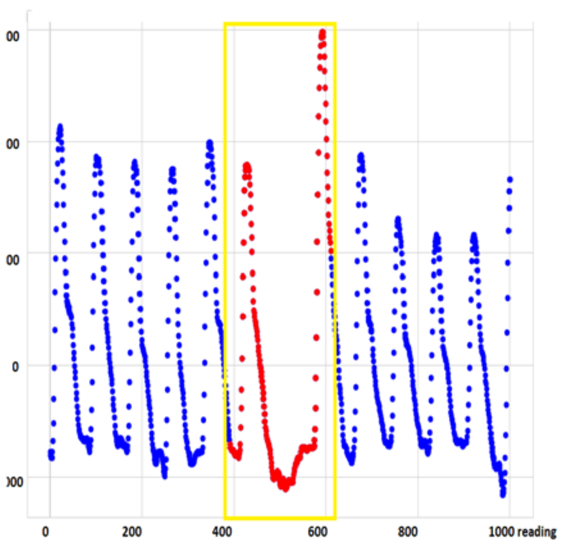


Figure 7.d TTDD output

Figure 7 implementation of IF, EIF and TTDD on the UCR_Anomaly dataset

DOI: 10.21608/ERJENG.2023.190373.1148

Journal of Engineering Research (ERJ)
Vol. 7 – No. 1, 2023
©Tanta University, Faculty of Engineering
https://erjeng.journals.ekb.eg/
ISSN: 2356-9441
e ISSN: 2735-4873

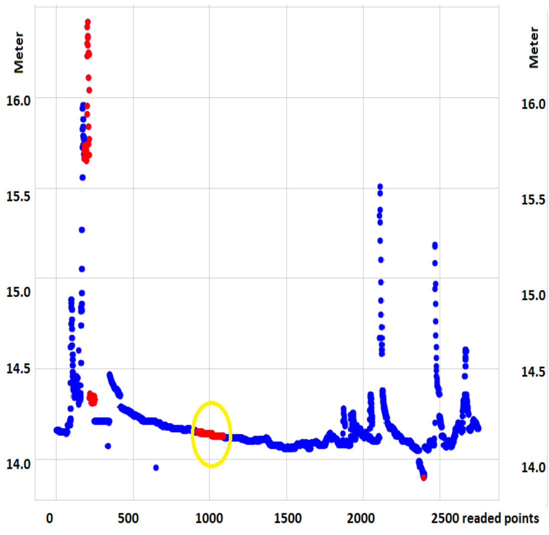


Figure 8.a Oddwater test set

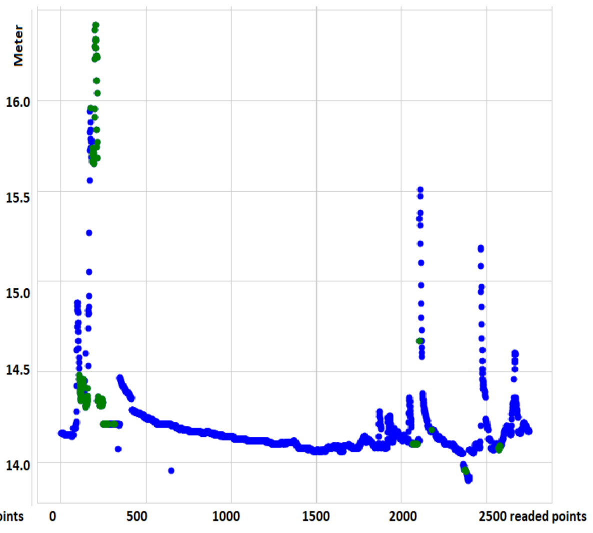


Figure 8.b IF output

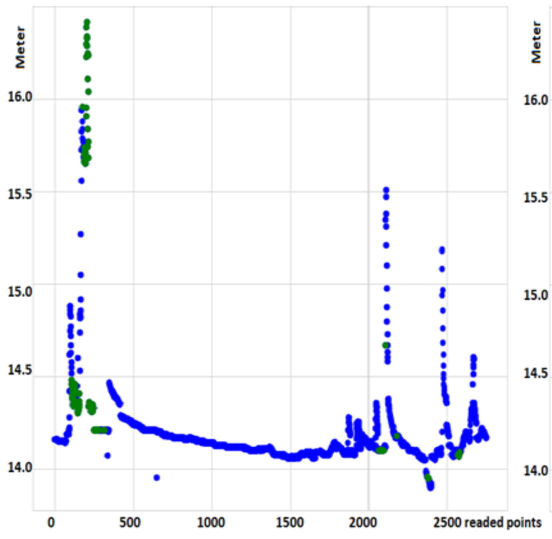


Figure 8.c EIF output

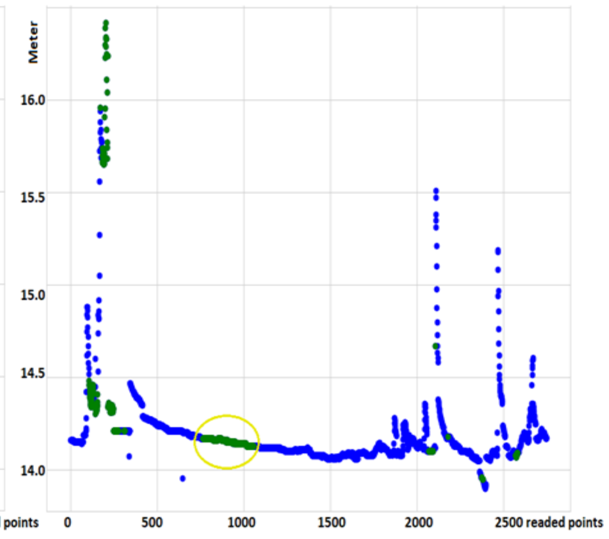


Figure 8.d TTDD output

Figure 8. Implementation of IF, EIF and TTDD on the Oddwater dataset

V. CONCLUSIONS

In this paper, we introduced a five functions framework to test, transform, decompose, and detect the outlier in the time series datasets. The framework starts to investigate the dataset from the Therefore, TTDD does not call the decomposition and Fourier Transformation functions and is the outputs for the datasets are summarized in table 4. We compared our proposed framework with the IF test stage to detect outlier stage. To this end, the proposed framework comprises main components; each task is achieved by its associated component.

The framework is mainly based on EIF algorithm that has the ability to detect the outlier with high accuracy rate. To validate the performance of the proposed framework, we conducted three time series datasets for experiments. Experimental results showed that the framework is accurate, effective and has the highest accuracy rate compared to related work.

In the future, we plan to develop and investigate the EIF used function, as it depends on trial and error in choosing the number of ntrees and the number of sample_size, and this has a significant impact on the results. Due to the importance of this step, as it directly affects the results, so it needs more and more research and development. Furthermore, we should investigate also the frequency comparison algorithm.

Funding: This research has not received any type offunding.

Conflicts of Interest: The authors declare that there is no conflict of interest.

Journal of Engineering Research (ERJ)
Vol. 7 – No. 1, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

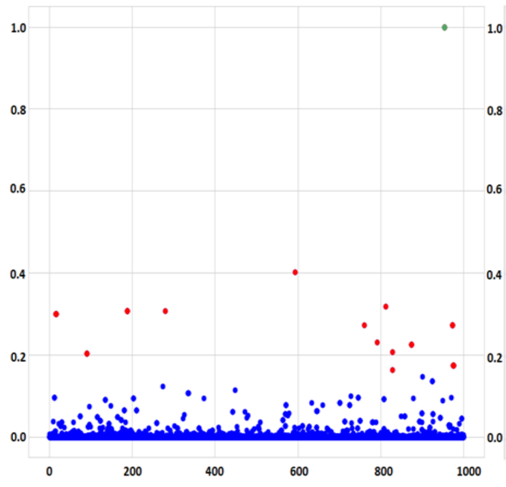


Figure 9.a SDWA test set

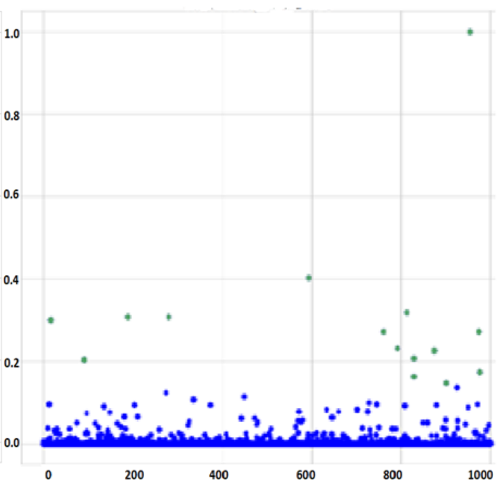


Figure 9.b IF, EIF and TTDD output

Table 3. TTDD each function's output

Dataset	Training set rows	Test set rows	Suitable period	EIF ntree	EIF Sample size	EIF Optimal threshold	F1 score			Accuracy score		
							IF	EIF	TTD D	IF	EIF	TTD D
UCR	2500	1000	320	300	26	0.6	0.44	0.44	0.95	0.8	0.8	0.97
	2500	1000	320	200	26	0.61	0.44	0.44	0.95	0.8	0.8	0.98
	2500	1000	320	150	15	0.59	0.44	0.44	0.95	0.8	0.8	0.98
	2500	1000	320	550	15	0.81	0.44	0.44	0.95	0.8	0.8	0.98
Oddwater	1842	4460	120	250	5	0.55	0.59	0.83	0.92	0.44	0.83	0.92
	4460	1842	120	270	5	0.52	0.59	0.83	0.89	0.44	0.83	0.89
	3629	2674	120	250	5	0.54	0.59	0.83	0.91	0.59	0.83	0.91
	3553	2750	200	445	5	0.53	0.59	0.87	0.91	0.59	0.87	0.91
	3550	2750	200	250	6	0.53	0.59	0.89	0.92	0.59	0.89	0.92
	2674	3629	250	250	5	0.53	0.59	0.86	0.86	0.59	0.86	0.90
SDWA	100000	10000	0	200	60	0.85	0.99	0.99	0.99	0.99	0.99	0.99

REFERENCES

[1] C.Aggarwal, Outlier Analysis, New York, USA: IBM T.J. Watson Research Center, 2017.
[2] L. Feremans, V. Vercruyssen, W. Meert, B. Cule and B. Goethals, "A Framework for Pattern Mining and Anomaly Detection in Multi-dimensional Time Series and Event Logs," in Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Würzburg, Germany, 2019.
[3] P. D. Talagala, R. J. Hyndman, C. Leigh, K. Mengersen and K. Smith-Miles, "A feature-based framework for detecting technical outliers in water-quality data from in situ sensors," ResearchGate, p. 32, 2019.
[4] Y. Yan, "Anomaly Detection for Water Quality Data," 2019.
[5] C.Aggarwal, Outlier Analysis, New York, USA: IBM T.J. Watson Research Center, 2017.
[6] S. Hariri, M. C. Kind and R. J. Brunner, "Extended Isolation Forest," Transactions on Knowledge and Data Engineering, p. 12, 2018.
[7] F. N. a. Z. W. b. R. W. b. a. X. L. Sisi Wang a, "Max–Min Robust Principal Component Analysis," Neurocomputing, vol. 521, pp. 89-98, 2023.
[8] UmaRao and P. V, "Time series decomposition model for accurate wind speed forecast," Prema and Rao Renewables, p. 11, 2015.
[9] M. G. a. A. Dengel, "Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm," in KI-2012: Poster and Demo Track, 2012.
[10] P. K. E. S. a. A. Z. Hans-Peter Kriegel, "Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data," in Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2009.
[11] N. B. a. F. C. Yahya Almardeny, "A Novel Outlier Detection Method for Multivariate Data," IEEE Transactions on Knowledge and Data Engineering , vol. 34, pp. 4052 - 4062, 2020.
[12] F. T. Liu, K. M. Ting and Z.-H. Zhou, "Isolation Forest," in International Conference on Data Mining, Pisa, Italy, 2008.
[13] A. M. Sarhan, "Data Mining in Internet of Things Systems A Literature Review,," Journal of Engineering Research (ERJ) , 2022..
[14] A.Emmott, S.Das, T.Dietterich, A.Fern and W.Wong, "A meta-analysis of anomaly detection problem," arXiv preprint arXiv:1503, 2015.
[15] H. WANG, M. J. BAH and M. HAMMAD, "Progress in Outlier DetectionTechniques: A Survey," IEEE Access, 2019.
[16] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data.," PLOS ONE , 2016.
[17] S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney and D. Song, "Anomalous Instance Detection in Deep Learning: A Survey.," arXiv preprint arXiv:, 2021.
[18] G. PANG, C. SHEN, L. CAO and A. V. D. HENGEL, "Deep Learning for Anomaly Detection: A Review," arXiv:2007.02500v3, 2020.
[19] K. M. Ting, S. Aryal and T. Washio, "Which Anomaly Detector should I use?," in IEEE International Conference on Data Mining, Singapore, 2018.
[20] V.Hodge and j.Austin, "A Survey of Outlier Detection Methodologies.," Kluwer Academic Publisher, p. 43, 2004.
[21] Y. Z. X. H. N. B. C. I. G. H. C. Zheng Li, "ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions," arXiv:2201, vol. 3, 2022.
[22] M. S. ,. a. A. Z. Hans-Peter Kriegel, "Angle-Based Outlier Detection in High-dimensional Data," ACM, pp. 444-452, 2008.

DOI: 10.21608/ERJENG.2023.190373.1148

Journal of Engineering Research (ERJ)
Vol. 7 – No. 1, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

[23] Z. Li, Y. Zhao, N. Botta, C. Ionescu and X. Hu, "COPOD: Copula-Based Outlier Detection," in IEEE International Conference on Data Mining (ICDM), 2020.
[24] S.-C. C. K. S. a. L. C. Mei-Ling Shyu, "A Novel Anomaly Detection Scheme Based on Principal Component Classifier," in IEEE Foundations and New Directions of Data Mining Workshop, 2003.
[25] B. P. J. S.-T. J. S. A. a. W. Scholkopf, "Estimating Support of a High-Dimensional Distribution," in Neural Computation, 2001.
[26] J. Z. ,. R. Q. ,. J. Y. a. Y. R. Junxiang Chen, "An Anomaly Detection Method for Wireless Sensor Networks Based on the Improved Isolation Forest," MDPI, 2023.
[27] H. Hoffmann, "Kernel PCA for novelty detection," in Pattern Recognition , 2007.
[28] R. A. V. N. G. A. B. a. E. M. Lukas Ruff, "Deep One-Class Classification," in the 35 th International Conference on Machine, Stockholm, Sweden, 2018.
[29] I. H. A. P. M. N. W. G. D. a. A. L. Christopher P. Burgess, "Understanding disentangling in β-VAE," in 31st Conference on Neural Information Processing Systems (NIPS), 2018.
[30] Z. L. C. Z. Y. J. J. S. M. W. a. X. H. Yezheng Liu, "Generative Adversarial Active Learning for Unsupervised Outlier Detection," in IEEE Transactions on Knowledge and Data Engineering, 2019.
[31] A. F. Hassan, S. Barakat and . A. Rezk, "Towards a deep learning-based outlier detection approach in the context of streaming data," Journal of Big Data, p. 16, 2022.
[32] C. Tao, "Applications of Bayesian Neural Networks in Outlier Detection," Big Data, 2023.
[33] K.Hundman, V.Constantinou, C.Laporte, I.Colwell and T.Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018.
[34] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong and Q. Zhang, "Time-Series Anomaly Detection Service at Microsoft.," in the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019.
[35] R. Yu, H. Qiu and a. Y. Liu, "A survey on social media anomaly detection," ACM SIGKDD Explorations Newsletter, pp. 1-14, 2016.
[36] Yu, X.He and Y.Liu, "GLAD: group anomaly detection in social media analysis.," ACM Transactions on Knowledge Discovery from Data (TKDD), 2015.
[37] Y. Djenouri and A. Zimek, "Outlier detection in urban traffic data," in the 8th International Conference on Web Intelligence, Mining and Semantics, 2018.
[38] D. Li, D. Chen, B. Jin, L. Shi, J. Goh and S.-K. Ng, "MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks.," arXiv preprint arXiv:1901.04997, 2019.
[39] M. R. C. S. F. B. L. a. V. R. C. Houssam Zenati, "Adversarially Learned Anomaly Detection," in IEEE International conference on data mining (ICDM), 2018.
[40] J.Chen, S.Sathe, C.Aggarwal and Turaga, "Outlier detection with autoencoder ensembles.," in SIAM International Conference on Data Mining, 2017.
[41] D. N. Gujarati and D. C. Porter, Essentials of Econometrics 4th Edition, 2021.
[42] Holloway and Nicholas, "https://www.kaggle.com/nholloway/stationarity-smoothing-and-seasonality," 2018. [Online].
[43] Hyndman and J.Rob, "Transformations, stationarity, differencing," in Forecasting:Principles and Practice, 2021.
[44] J. Perktold, S. Seabold and J. Taylor, "https://www.statsmodels.org/stable/examples/notebooks/generated/stationarity_detrending_adf_kpss.html#KPSS-test," 2019. [Online].
[45] U. Oberst, "The Fast Fourier Transform," SIAM Journal on Control and Optimization, 2007.
[46] J. Perktold, S. Seabold, J. Taylor and statsmodels-developers, "https://www.statsmodels.org/dev/generated/statsmodels.tsa.seasonal.seasonal_decompose.html," 2019. [Online].
[47] s.-l. developers, "https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html," Sphinx-Gallery, 2020. [Online].
[48]"https://numpy.org/doc/stable/reference/generated/numpy.fft.fftfreq.html," NumPy, 2008-2022. [Online].
[49] "https://echo.epa.gov/tools/data-downloads#dwdownloads," United States Environmental Protection Agency, 1 2017. [Online]. [Accessed 7 2021].
[50] T. Fawcett and H. Laboratories, "ROC Graphs: Notes and Practical Considerations for Researchers," Kluwer Academic Publishers., 2004.
[51] Sunkara, "https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/," 12 2020. [Online]. [Accessed 7 2021].
[52] J. Brownlee, "https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/," 2021. [Online]. [Accessed 7 2021].
[53] B. Rosner, "Percentage Points for a Generalized ESD Many-Outlier Procedure," echnometrics 25, p. 165–172., 1983.
[54] R. B. R. M. P. a. P. A. Paternoster, "Using the correct statistical test for the equality of regression coefficients.," Criminology, Vols. 859-866, p. 36, 1998.
[55] Y. Z. X. H. N. B. C. I. G. H. C. Zheng Li, "ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions," arXiv:2201, vol. 3, 2000.
[56] A. B. V. K. VARUN CHANDOLA, " Anomaly detection: A survey," ACM computing surveys, vol. 41, p. 15, 2009.