Hindawi
Security and Communication Networks
Volume 2021, Article ID 6677027, 10 pages
https://doi.org/10.1155/2021/6677027

Research Article

An Anomaly Detection Algorithm Selection Service for IoT Stream Data Based on Tsfresh Tool and Genetic Algorithm

Zhongguo Yang[1], **Irshad Ahmed Abbasi**[2], **Elfatih Elmubarak Mustafa**[2], **Sikandar Ali**[3,4] **and Mingzhu Zhang**[1]

[1]*School of Information Science and Technology, Beijing Key Laboratory on Integration and Analysis of Large-Scale Stream Data, North China University of Technology Beijing, Beijing, China*
[2]*Department of Computer Science, Faculty of Science and Arts at Belgarn, P.O. Box 60, Sabt Al-Alaya 61985, University of Bisha, Saudi Arabia*
[3]*Department of Computer Science and Technology, China University of Petroleum-Beijing, Beijing 102249, China*
[4]*Beijing Key Laboratory of Petroleum Data Mining, China University of Petroleum-Beijing, Beijing 102249, China*

Correspondence should be addressed to Sikandar Ali; sikandar@cup.edu.cn

Received 31 December 2020; Revised 21 January 2021; Accepted 23 January 2021; Published 8 February 2021

Academic Editor: Shah Nazir

Anomaly detection algorithms (ADA) have been widely used as services in many maintenance monitoring platforms. However, there are numerous algorithms that could be applied to these fast changing stream data. Furthermore, in IoT stream data due to its dynamic nature, the phenomena of conception drift happened. Therefore, it is a challenging task to choose a suitable anomaly detection service (ADS) in real time. For accurate online anomalous data detection, this paper developed a service selection method to select and configure ADS at run-time. Initially, a time-series feature extractor (Tsfresh) and a genetic algorithm-based feature selection method are applied to swiftly extract dominant features which act as representation for the stream data patterns. Additionally, stream data and various efficient algorithms are collected as our historical data. A fast classification model based on XGBoost is trained to record stream data features to detect appropriate ADS dynamically at run-time. These methods help to choose suitable service and their respective configuration based on the patterns of stream data. The features used to describe and reflect time-series data's intrinsic characteristics are the main success factor in our framework. Consequently, experiments are conducted to evaluate the effectiveness of features closed by genetic algorithm. Experimentations on both artificial and real datasets demonstrate that the accuracy of our proposed method outperforms various advanced approaches and can choose appropriate service in different scenarios efficiently.

1. Introduction

With the growth of the Internet of Things (IoT), the sensor or stream data is bound to be collected at tremendous speed. In such real-time scenarios, there can be various anomalous data streams, for example, the data diverge from the usual behavior of the stream or the abruptly jumped data [1], which are dissimilar to familiar patterns.

It is critical for further decision making to capture these anomalous data accurately and timely. Banerjee et al. [2] introduced the trend of everything as a service (XaaS). Following Banerjee et al. [2], a lot of researchers try to encapsulate various data or common functions into services. For example, streaming as a service is studied by many researchers [3–5], which can provide the sharing and simple processing capabilities for stream data. The idea of choosing suitable service or methods can be referred to in [6–8]. It is proposed to provide common functions for various data sources, which enable users to conveniently reuse these functions and form more complex functions through service composition.

In real-world software systems, numerous anomaly detection algorithms (ADAs) are industrialised and are offered as a service to be utilised in diverse domains [9, 10]. In our preceding work [11], a proactive data services abstraction was applied to appropriately encapsulate present ADAs into a service.

Even though, with the scenario in hand, it is still a challenge to effectively capture anomalous data considering various circumstances. Following the concept of the No-Free-Lunch (NFL) optimisation theorem [12], it is infeasible to find a single algorithm for all the cases that dominate all others on the same optimisation problem [1]. In the state-of-the-art survey paper, Braei and Wagner [13] state that for the most part the univariate dataset may suffer from contextual anomalies; therefore, statistical methods will not perform well. Deep learning models may perhaps increase the area under the curve (AUC) and neural network models might outperform the statistical methods. On contrary, the volume of novel stream data can appear frequently and continuously and can result in missing part of the anomalous data through manual service selection. Consequently, running an ADS possibly will not adjust to different types of stream data. Therefore, for faster and more accurate anomaly detection, it is obligatory to choose an appropriate service for different stream data dynamically at run-time.

Since each type of anomaly detection algorithms gives better results only for a particular set of stream data [14]. Therefore, to automatically choose appropriate services for diverse IoT scenarios, it is required to correctly and quickly characterise the underlying stream data. Hence, proper service might be chosen and configured based on the pattern of a particular stream of data. Keeping in view the gigantic volume of stream data, this study finds out that several IoT streams are alike owing to their shape similarities and implicit relations.

For effective handling of anomalies from various stream data, based on the above observation, in this paper, an Anomaly Detection via Service Selection (ADSS) framework was proposed. To recognise the pattern of various stream data, in our proposed ADSS framework, it tries to capture intrinsic similarity and dissimilarity in various stream data established on time-series statistical features. Moreover, a fast classifier based on the XGBoost algorithm is trained to record features of stream data in order to detect appropriate ADS dynamically at run-time. Due to the presence of the best classifier, our ADSS method can identify the dynamics of data stream patterns of newly appearing stream of data and then choose and configure the suitable service.

Firstly, it is well known that there could not be an algorithm that could defeat others in all the datasets. Consequently, our aim of this study is not to build a model or to develop a new algorithm which could beat all the other algorithms in all the datasets. Instead, a method is designed to capture the variation of the stream data in the run-time and configure different algorithm to handle the stream data. Experimental results show that we could achieve a better performance in the long run.

This study focuses on the selecting algorithms based for dynamically changing IoT stream data. The original idea is to construct features to be a representation of different stream data and build a supervised model to recommend a suitable algorithm for a certain stream data. Collections of historical data are gathered from a real monitor system. Further, on the basis of these data, an XGBoost model is trained based on the feature and its label. Here, the label is the best algorithm which is more suitable for a certain kind of stream data. This manuscript is the extended version of our recently published conference paper [15].

In the revised version of the manuscript, the features' construction process is improved by applying a Tsfresh tool and intelligent optimisation algorithm [16]. The former tool is taken to extract multiple features of time-series. These features consist of 100+ kinds of features from different angles which could represent intrinsic features of stream data completely. Moreover, an intelligent optimisation algorithm such as genetic algorithm is applied to help choose a subset of features which could further result in the reduction of computing complexity of the algorithm recommendation procedure. The specific contributions of the manuscript are summarised below:

(i) In this paper, we develop a method that facilitates IoT-based systems to automatically choose appropriate services using the existing data features in order to detect an anomaly.

(ii) In this paper, we develop a service update framework in which service quality and its resultant algorithms and data stream are recorded. The aforesaid historic data will assist in the training of different decision models that paves the approach for accurately recommending ADS. In this approach, freshly designed algorithms can easily be added to the service pool.

(iii) In this paper, we carried out various experiments by means of data streams from NAB [17] and Yahoo datasets [18]. The experimental results demonstrate that our method can select the best service dynamically according to changes in the stream data pattern.

(iv) In this paper, an improved features' construction method by applying Tsfresh tool and intelligent optimisation algorithm is devised [16].

The remainder of this article is organised as follows. Section 2 describes the related work to build a proper problem statement. The proposed ADSS framework is accessible in Section 3, while Section 4 is based on experimental outcomes. Section 5 is the last section which summarises the paper.

2. Background and Related Work

2.1. Anomaly Detection Algorithms. In this study, the unsupervised methods are mainly considered to detect anomalies due to its good generalization ability. The possible reasons why we do not consider supervised methods are as follows. Firstly, in real-time IoT arrangements, different types of time-series data are collected that are hard to label for anomalies. Secondly, to rapidly deploy ADS, almost there is very less or no time to train a complex anomaly detection model. Thirdly, for the dynamic change of time-series in real-time IoT systems, even some of the good models perform badly and cannot handle this dynamism. Summary of the unsupervised class of ADAs is given in Table 1.

Table 1: Summary of stream data anomaly detection algorithms.

Typical algorithms	Category	Characteristic and limitations
Prediction confidence interval (PCI) for time-series outlier detection, simple exponential smoothing (SES) [19], and ARIMA model [20]	Statistical approaches	(1) A supposition about outlier data and normal data need to made first (2) Domain-specific knowledge is needed for threshold selection depends on
Autoencoder [21], LSTM [22]	Artificial neural computing	Since clustering methods cannot deal with continuous changes in data, therefore careful parameter tuning is needed
Density-based spatial clustering of applications with noise (DBSCAN) [23], subsequence time-series clustering (STSC) [13], isolation forest [24], local outlier factor (LOF) [25], one-class support vector machine (OC-SVM) [26]	Machine learning approaches	Work on stream data; therefore, the normal reference model might be outdated at the moment they are actually used

Although, for anomaly detection, there are numerous deep learning algorithm-based methods, for example, AutoEncoder [21] and LSTM [22], they cannot be used directly on the continuous stream data, because these methods need fine parameter tuning and a lot of training data. Allowing for the scenario of frequent changes of data pattern or anticipated behavior in the frequently launched streams stream, the notion of selecting appropriate service algorithms is becoming challenging.

2.2. Anomaly Detection Algorithm in the System. To present a unified and easier way to adapt to changes and accurately detect anomalies in diverse circumstances, a lot of ADAs are delivered as a service.

In [14], the first ever ADS framework was developed to consider the aforementioned problems through semi-supervised learning and clustering. This study was the first work that applies semisupervised learning to key performance indicator (KPI) anomaly detection [14]. Still, the postulation of huge resemblance in KPI stream data is not effective in conventional IoT stream data.

In [10], an anomalous behavior recognition system composed of two phases was developed based on the past data learning the normal behavior of the system in the first phase and then by processing real-time data and detecting abnormal behavior in the system dynamically in real time in the second phase. In their system, complex event processing (CEP) patterns and anomaly detection are combined as a REST service to be utilised through the interface by a user.

In [27], the authors divided stream data into four different time-series groups, i.e., periodic, stationary, non-periodic, and nonstationary. Furthermore, they used diverse techniques to detect anomalous data.

In [28], the authors state that, in the age of big data, it is a very challenging but important task to detect anomalies. They presented the Interactive Data Exploration As-a-Service approach for the identification of significant data.

A dynamic IoT stream data ADA must recognise various data pattern changes in diverse stream data anomaly detection approach. Though previously researchers were aware of the problem of runtime outlier detection, yet solution formation did not consider this problem and ignored consequent changes in the stream data. While working with a fast growing volume of IoT data with their respective dynamic nature, current approaches are not effective.

We attempt to develop a framework based on the features collected in the first phase to characterise the time-series data and then apply deep learning models in the second phase to recognise the pattern of data that will help reconfigure the ADS dynamically in the run-time.

3. Framework for IoT Stream Data ADS Selection

3.1. Description of Our Proposed ADS Framework. The framework developed in this paper comprises of three parts: (i) service selection procedure, (ii) encapsulations of ADAs, and (iii) service applied procedure. Many publicly available unsupervised ADAs are incorporated for the development of ADAs. As mentioned before, the available ADAs can be encapsulated into services based on PD-service abstraction. A RESTful API is used for the selection of ADS. Service receiver can define individual views to build IoT applications and can get the anomalous data via the Uniform Resource Identifier (URI) of service. The entire working of the developed ADSS framework is illustrated in Figure 1. As shown in Figure 1, the collected tuples are portions of historical data that can be collected through recording the stream data for a long time by field experts along with the appropriate ADAs.

Stream data along with its appropriate algorithm are kept in the database in the form of a tuple **<stream data, algorithm>** that can be used as a metadata for onward service identification and selection procedure. These historic data can be updated by collecting running examples from anomaly detection systems or by experts in this field. Usually, these recorded data monitored the performance of various ADAs and stream data that can be used to generate a scheme to select an ADS for specific stream data.

ADSS is the basic unit of our framework. Each stream data can be represented by a feature vector for by applying a stream feature extraction technique on stream data. As a training data of service selection model, the paired data are constructed and combined with the best possible service. In the service applying part of our framework, a new stream data is transformed to features vector grounded on the same

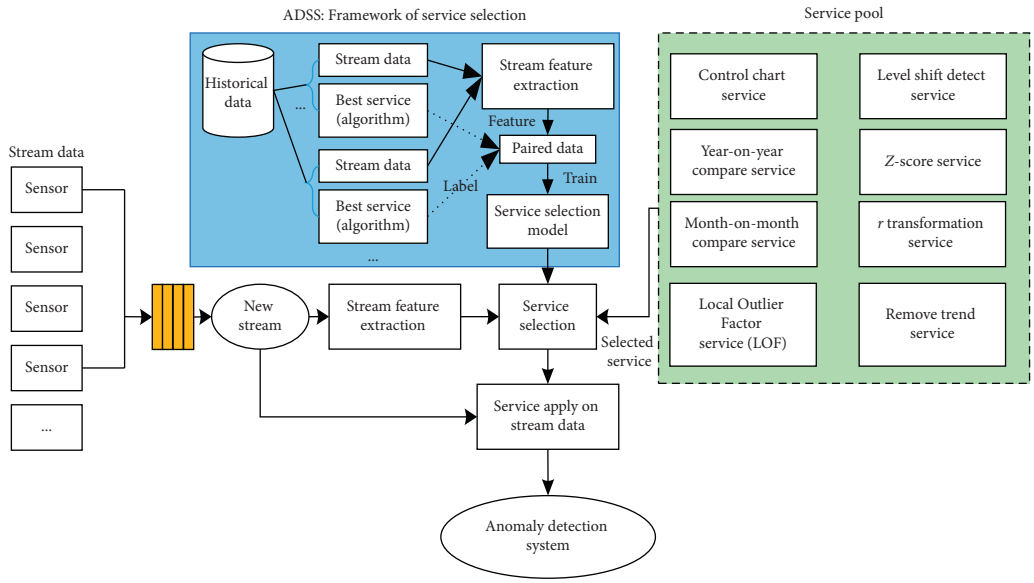

Figure 1: Anomaly detection via service selection framework for service selection.

feature extraction technique. Finally, the service selection model is used on the feature vector to select appropriate service for the existing stream data and ultimately call the service in real time to identify anomalous data.

3.2. Model for Service Selection. Abundant stream data are gathered and their feature is extracted through the process discussed in the previous sections in order to select appropriate services for stream data anomaly detection. The finest ADS is chosen by analysing the historic data based on the recorded stream data fragment and its corresponding best service. In general, few ordinary services are tested repeatedly on these stream data fragments to identify its finest service. Grounded on the stream data fragments and its finest ADS, the service selection problem has been transformed into a pattern recognition problem.

Taking into account its computing efficiency, in this paper, XGBoost [29] is utilised as a base classifier to choose a service for real-time stream data anomaly detection. This procedure is illustrated in Figure 2. It should be noted that any classifier can be used in our framework. However, in this study, we have chosen the XGBoost algorithm as to best choose service considering the easy explanation and high computing efficiency of the XGBoost algorithm.

The time-series features are the main part of our framework, as presented in Figure 2. Some renowned stream data features are taken from publicly available features and some former anomaly detection schemes. What is more, a feature selection method was employed to find some good features to capture stream data essential features. The objective of the selected features of stream data is to accurately and quickly select the appropriate service dynamically in real time for novel evolving stream data.

3.3. Stream Data Patterns Representations. Stream data may generate dissimilar patterns as demonstrated in Figure 3. According to Bu et al. [14], supervised techniques such as SVM or deep learning-based techniques are not achievable for the huge amount of novel IoT stream data applications and the dynamic nature of the stream data. This might be due to two reasons: difficult parameters tuning process and a large amount of training data.

Researchers like Bu et al. [14] state that for some kinds of stream data simple ADAs may perform well compared to some multifaceted algorithms such as deep learning. The pattern of stream data can also be recognised in time which overlays the way for future algorithm selection in modern microservice architecture also recognised as a service selection. The main contribution of our work focuses on the extraction of features to characterise stream data and based on these features select suitable algorithm service.

In order to select useful features that could distinguish different stream data patterns, a feature selection method was applied. We surveyed all the features which could be considered for the representation of time-series data. There are multiple types of features from different angles such as statistics, mathematics, shape, distribution of data, and others in the classification of time-series field.

Christ et al. [16] automatically extract 100 features from time series and develop a tool called Tsfresh. These features label basic characteristics of the time series, for example, maximal or average value, the number of peaks, and additional complex features, for example, time setback symmetry statistics. At the same time, through hypothesis testing to reduce the characteristics to those, which can best explain the trend called decorrelation. These feature sets are then used to construct machine learning or statistical models based on time series data such as classification or regression tasks.

In addition, these collected features are the reflection of the inherent nature of data patterns, for example, the distribution, the fluctuation, and shape of data. Some typical features are demonstrated in Figure 4.

As is shown in Figure 4, these simple features or complex features are designed to characterise the time-series data

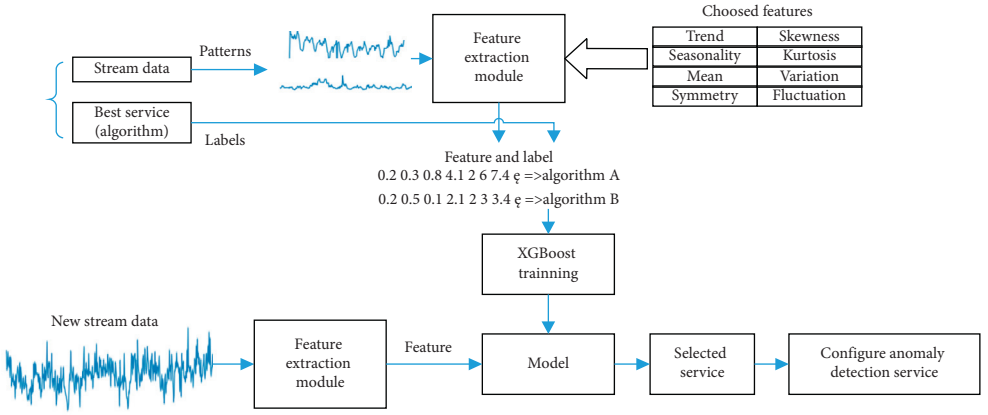

Figure 2: The process of service selection based on feature extraction.

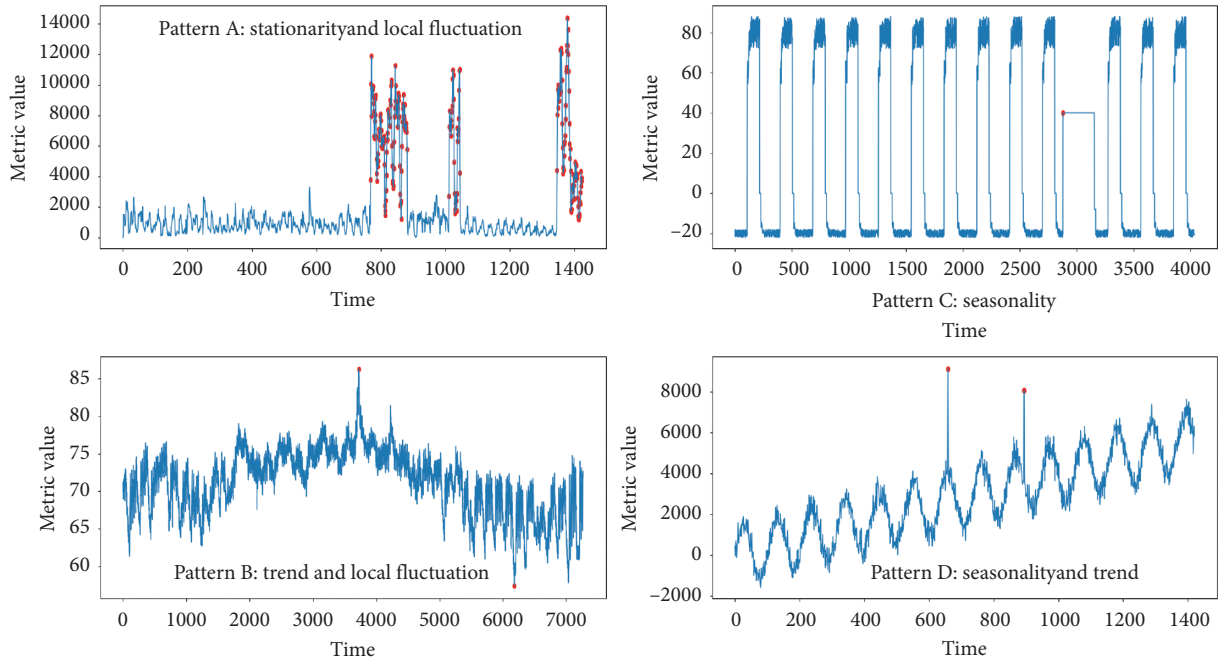

Figure 3: Behavior characteristic of time-series in real-world sensor datasets. Anomalous data characteristics are signposted in red. The time-series data are taken from Yahoo! datasets [18] and NAB [17].

from different angles and own their special geometric interpretation or statistical meaning. These special characteristics are quantified by computing these features. In other words, it is possible to distinguish these stream-data from each other by comparing these features. More details about other features in Tsfresh are discussed in Table 2.

As is presented in Table 2, some computing techniques are taken from Extendible Generic Anomaly Detection System (EGADS) [30], and some metrics are taken from Tsfresh [16] and the rest from other renowned statistical techniques such as standard deviation and mean. Local fluctuation, metrics of symmetrical values, and fluctuation ratio are recommended in our study to characterise stream data from diverse perspectives.

The flowchart of selecting features from multiple original features is illustrated in Figure 5. As shown in Figure 5, the genetic algorithm (GA) [31] is applied to find a feature subset, which is enough to characterise different traits of various stream data.

In the process of GA, the fitness computing consists of two steps: decoding individual to feature subset and computing test score based on the feature subset. The test score is utilised as the fitness of the individual. The other steps of GA such as selection, crossover, and mutation are following the normal behavior as in the traditional computing processes.

The above process belongs to wrapping feature selection approaches which build many models with dissimilar subsets of input features and hand-picked those features that have best performance agreeing to the performance metric. Although these approaches are independent of the types of variables, yet they might be computationally expensive.

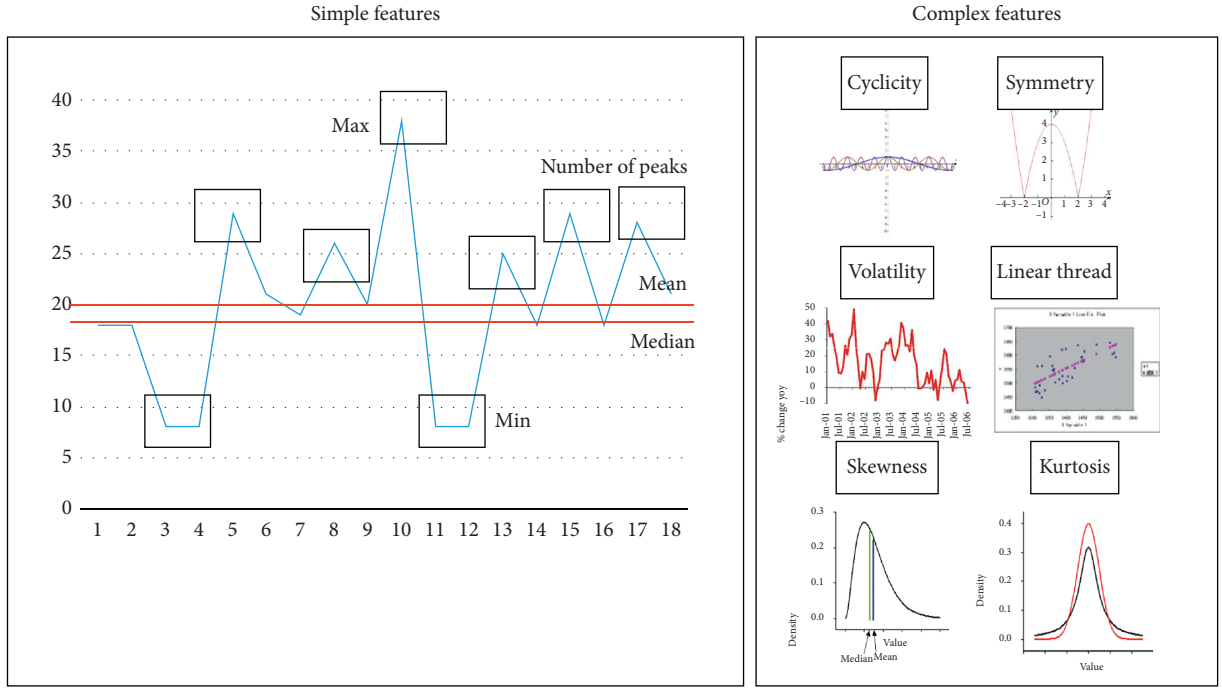

Figure 4: The meaning of features in stream data.

Table 2: The name, design principle, and computing method of some features in Tsfresh.

Name	Design principle	Computing method [16, 30]
Mean	The baseline of time series	$\overline{x} = \text{mean}(x_{t-w}: x_{t+w})$
Standard deviation	The standard deviation of time series	$\text{std} = \sqrt{\sum_{i=1}^{N}(x_i - \overline{x})^2/N}$
Coefficient of variation	The reflection of the degree of data dispersion	$\text{cv} = \text{std}/\text{mean}$
Local fluctuation 1	The difference of the smooth curve and original curve	$s_{\text{diff}} = \frac{1}{n}\sum_{i=1}^{n}\lvert x_i - x_i^*\rvert$
Local fluctuation 2	Local fluctuation with a dynamic step	$d(\text{step}) = (1/2\text{step})\sum_{i=1}^{n-\text{step}}(x_i - x_{i+\text{step}})^2$
Smooth factor	The ratio of the whole number to the number of turning points	$s_{\text{smooth}} = (1/n-2)N_{\text{change}}$
Symmetrical value	The symmetry of the curve	$\text{sym} = \sum_{i=1}^{n/2} x_i / \sum_{i=n/2}^{n} x_i$
Fluctuation ratio	Whole fluctuation power	$\text{quantil}(x_{\text{norm}}, 0.9) - \text{quantil}(x_{\text{normal}}, 0.1)$
Skewness	The estimation of the degree of statistical data distribution and the direction of skew is the digital characteristics of the asymmetric degree of statistical data distribution	$S = (\sum_{i=1}^{n}(x_i - \mu)^3/n\sigma^3)$
approximate_entropy	Approximate entropy is used to measure the periodicity, unpredictability, and volatility of a time series	Refer to [16]
Autoregressive coefficient	Measure the cyclical nature of data	$\frac{1}{n-1}\sum_{i=1,\ldots,n} 1/(n-l)\sigma^2\sum_{t=1}^{n-l}(X_t - \mu)(X_{t+l} - \mu)$
Kurtosis	The feature number indicating the peak value of the probability density distribution curve at the average value	$E[(X - \mu/\sigma)^4]$
absolute_sum_of_changes	Absolute sum of first-order difference	$\sum_{i=1}^{n-1}\lvert x_{i+1} - x_i\rvert$
Linear_trend	Calculation of a linear least squares regression for the values of the time series to the sequence from 0 to the length of the time series −1	Refer to [16]
fft_aggregated	Returns the variance, mean, kurtosis, skewness, and absolute Fourier transform spectrum	Refer to [16]

Though these features are designed for general classification and clustering problems, and not for algorithm selection problems, as dictated by the literature on machine learning technology, the transform learning technology may perform well in similar problems for various problem fields. Considering the similarity of the above two problems, a conclusion can be drawn that the selected features are useful in the algorithm selection task.

Finally, these features will help choose a suitable algorithm for a certainly given stream of data by training a classification model. As it is mentioned before, if these features could be computed in real time, the decision of

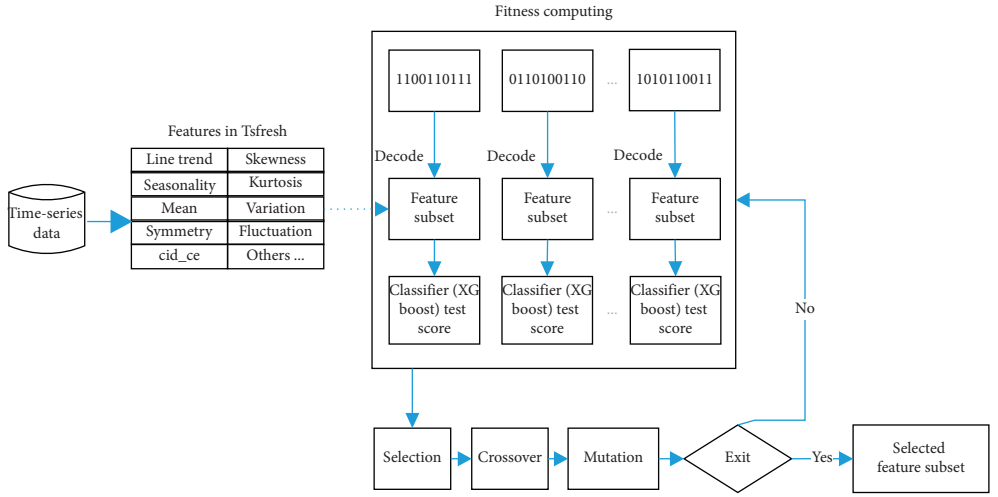

Figure 5: The process of selecting features from Tsfresh features.

choosing the optimum algorithm service will be quicker and thus it will be accepted by many application users.

4. Experimental Validation and Interpretation

4.1. Datasets. For the introduction and assessment of univariate methods, several time-series datasets as listed in Table 3 have been selected. As is presented in Table 3, the reason for selecting these datasets for the assessment of the proposed framework is the availability of similar characteristics in the data. The synthetic and real data encompasses all the commonly known three anomaly forms: random, collective, and point anomaly [14].

4.2. Preprocessing of Data. Standardisation helps numerous machine learning approaches to converge quickly. A dataset is said to be standardised one if its standard deviation σ is 1 and its mean μ is 0. Mathematically, let D be the dataset and σ the standard deviation of D while μ is its mean. Then, standardised D is given by the following equation:

$$\hat{x} = \frac{x - \mu}{\sigma}, \quad \forall x \in D. \tag{1}$$

4.3. Metrics Evaluation. The performance of our developed framework is evaluated by plotting the receiver operating characteristic (ROC) curve. As a first step, False Positive Rate (FPR) and True Positive Rate (TRP) are illustrated below:

$$\begin{aligned} \text{FPR} &= \frac{FP}{P}, \\ \text{TRP} &= \frac{TP}{P}, \end{aligned} \tag{2}$$

where FP denotes the total number of wrong positive predictions, TP denotes the total number of correct positive predictions, and P is the total number of positive-labeled values. A list of $\delta \in R$ are used as a threshold that leads to various pairs of FPR and TPR for each δ. A list of two-dimensional coordinates from values already computed is made, and then they will be plotted as a curve. The starting pair of points for this curve will be (0, 0) while the ending pair of points will be (1, 1), respectively. The area under the curve is labeled as AUC. Higher AUC represents the higher possibility that the dignified algorithm allocates anomalous points randomly to the time series. Furthermore, higher anomaly scores than random normal points will enable AUC to correctly associate with various anomaly detection approaches. Thus, in this study, AUC is chosen as an evaluation metric.

4.4. Comparison of Various Methods. Five algorithms out of numerous sets of algorithms such as Long Short-Term Memory Networks (LSTM) [22], Local Outlier Factor (LOF) [25], Prediction Confidence Interval (PCI) [20], One-Class Support Vector Machines (OC-SVM) [26], and Autoencoder [21] are set as baseline algorithms. These algorithms represent machine learning techniques, deep learning techniques, and statistical techniques that are developed for anomaly detection in stream data. Some of the hyperparameters used in our study are borrowed from the work of Bu et al. [14]. Table 4 explains the hyperparameters of these algorithms.

4.5. Experimental Procedure and Outcome Analysis. First, each and every dataset is divided into training set 60% set and testing set 40% using a stratified statistical sampling technique. Each time series of the training dataset and its appropriate algorithm are constructed and computed as a paired dataset. Secondly, the XGBoost model is trained to recognise the patterns of a stream using the paired dataset as an input. Thirdly, the trained XGBoost model is used for the recognition of patterns in each time-series in the test set and finds out a suitable algorithm as a service. Finally, the performance of ADS with the recommended algorithm employed on each time series is evaluated.

The AUC values of the anomaly detection datasets are presented in Table 5. The outcomes presented in Table 5

Table 3: The datasets used in our experiment.

Name	Source	Number of time-series	Number of time stamps	Ratio of anomalous data (%)	Characteristic
Dataset 1	Yahoo [18]	100	1680	0.5	Artificial univariate time-series data comprises of anomalies' change point where it changes the mean of the time series
Dataset 2	Yahoo [18]	100	1680	0.3	Artificial univariate time-series data with anomalies and seasonality are introduced at random points
Dataset 3	Yahoo [18]	100	1421	0.3	Artificial univariate time-series data
Dataset 4	Yahoo [18]	67	1420	1.9	A univariate Yahoo services time-series dataset recording the traffic in which anomalies are by-hand pigeonholed. Majority of the time-series are static
NYCT	NAB [17]	1	10320	0.05	A univariate New York City taxi request time-series dataset comprising the New York City (NYC) taxi demand from July 1, 2014, to January 31, 2015, with an observation of the no. of passengers noted down every half hour. It comprises five shared anomalies that arise in the NYC: Christmas, thanksgiving, marathon, snowstorm, and New Year's Day.

Table 4: Description of our experimental datasets.

Model	Hyperparameter	Value
LOF	Distance function (k)	10, Minkowski distance
PCI	k, α	30, 98.5
LSTM	Filters, optimisers, architecture, loss, batch size, and epochs	$4 * 4$, Adam, 2-state full LSTM layer, MSE, 32, 50
OC-SVM	Upper bound of outliers, kernel	Radial basis function kernel (RBF), 0.7
Autoencoder	Architecture, activation functions, optimiser, loss, batch size, and epochs	Decoding layers (16, 32), encoding layers (32, 16), linear for output, ReLU for encoding and decoding, MSE, Adam, 32, 50

Table 5: The AUC values computed for each experimental dataset using our developed ADSS framework.

Time series	OC-SVM	PCI	LSTM	LOF	Autoencoder	ADSS
Dataset 1	0.939	0.689	0.589	0.952	0.597	0.955
Dataset 2	0.957	0.674	0.578	0.951	0.602	0.956
Dataset 3	0.995	0.762	0.734	0.995	0.743	0.995
Dataset 4	0.851	0.522	0.812	0.814	0.782	0.856
NYCT	0.586	0.54	0.841	0.493	0.697	0.841

proved that for a given dataset the most suitable algorithm may be different in each case. Results presented in Table 5 signpost that LSTM performs best for the NYCT dataset, LOF performs best for dataset 1, while OC-SVM achieves best for datasets 3 and 4. As is given in Table 5, out of the five datasets, our framework shows better performance in four. Even in the case of dataset 2, the performance is nearly equal to the OC-SVM which is the best algorithm. This is the reason; our ADSS framework for algorithm service selection can quickly and flexibly choose the most appropriate algorithm service for any type of data flow processing.

4.6. NAB Dataset and Its Outcome Analysis. In paper [32], researchers compared multiple anomaly detectors such as Skyline, Relative Entropy, and HTM-based algorithms. From its public available experiment reports, we found that Numenta algorithm could achieve the best average performance on all the datasets. However, for one certain dataset such as Twitter_volume_UP, the EarthgeckoSkyline could defeat other detectors. Inspired by the ensemble learning and algorithm selection strategy, we use the supervised learning method to choose a suitable detector for one certain dataset, so we show the experiment on NAB dataset. In NAB results, the evaluation metrics are Standard Score, Reward Low FP rate scores, and Reward low FN rate scores; for more information, one can refer to [17].

The process of the experiment is the same as that explained in Section 5; the performance of the experiments on the NAB dataset is shown in Table 6. As is demonstrated in Table 6, our framework had achieved better performance considering all these detectors as candidate ADAs. A conclusion can be drawn that our framework could recognise the feature of streaming data and help choose a good detector for it and achieve better performance on average.

4.7. Outcome Analysis. In our framework, the algorithm is decided and recommended as best for current stream data and be configured to check the anomalous data. The base algorithms can be added as needed and the available algorithms will become more and more. So, in the long run, when we add enough algorithms to the service pool, the final anomaly detection performance will become better. This framework takes full advantage of metalearning idea which recognises the stream data pattern and configures its best algorithm.

Table 6: The three kinds of scores for the NAB dataset.

Detector	Standard scores	Reward low FP rate scores	Reward low FN rate scores
Numenta HTM using NuPIC v0.5.6∗	70.1	63.1	74.3
Numenta™ HTM∗	64.6	56.7	69.2
htm.core	63.1	58.8	66.2
EarthgeckoSkyline	58.2	46.2	63.9
Relative entropy	54.6	47.6	58.8
Recommended by our framework	**77.9**	**68.1**	**83.2**

Our framework is not creating a new algorithm; instead it is choosing the finest algorithm for any time-series data, thus possibly improving the total performance of the entire IoT system. In general, our framework modifies the quality of service, in the background of encapsulation of algorithm as web service.

5. Conclusion

In practice, it is unfeasible to build a universal method to detect all types of anomalies in IoT stream data; we attempt to discriminate the data pattern and adjust appropriate ADS. Various ADSs can be chosen and then according to their stream data pattern, they can be configured. We attempt to extract features of a stream and select an appropriate algorithm for its anomaly detection.

Experimentations through five datasets (illustrated in Table 3) demonstrate the performance of our method and are presented in Table 5. The experimental outcomes described in Table 5 prove that our method is able to select the accurate service proficiently and can recognise the data pattern efficiently. Moreover, the result on the NAB dataset is shown to further illustrate the good performance achieved by our method.

To further analyse the experimental result, we found that our method is like an ensemble learning process that will merge together different kinds of models in order to achieve better results. Different from the traditional ensemble method, we try to capture the intrinsic characteristics of streaming data from the view of feature engineering. So, the Tsfresh tool and GA algorithm played an important role when finding the importance of features.

However, our method is able to select the service efficiently and can recognise the data pattern efficiently. Still, our method needs sufficient historical data to improve the accuracy of a service selection process that can be done by collecting further real-world data and experimenting with more artificial dataset in the future.

Data Availability

All the data used to report the findings in this paper are provided in the form of tables in the paper.

Conflicts of Interest

There are no conflicts of interest as declared by the authors of this paper.

Acknowledgments

This work was supported by Projects of International Cooperation and Exchanges NSFC (Grant no. 62061136006) and the Scientific Research Initiation Funds (Grant nos. 2462020YJRC001 and 110051360002).

References

[1] V. Chandola, V. Mithal, and V. Kumar, "Comparative evaluation of anomaly detection techniques for sequence data," in *Proceedings of Eighth IEEE International Conference on Data Mining*, pp. 743–748, Pisa, Italy, December 2008.

[2] P. Banerjee, R. Friedrich, C. Bash et al., "Everything as a service: powering the new information economy," *Computer*, vol. 44, no. 3, pp. 36–43, 2011.

[3] Q. Chen, M. Hsu, and H. Zeller, "Experience in continuous analytics as a service (CaaaS)," in *Proceedings of International Conference on Extending Database Technology*, ACM, Uppsala, Sweden, March 2011.

[4] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by internet of things," *Transactions on Emerging Telecommunications Technologies*, vol. 25, no. 1, pp. 81–93, 2014.

[5] Z. H. Ali, H. A. Ali, and M. M. Badawy, *A New Proposed the Internet of Things (IoT) Virtualization Framework Based on Sensor-As-A-Service Concept*, Wireless Personal Communications, Berlin, Germany, 2017.

[6] M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, "Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city," *Future Generation Computer Systems*, vol. 107, pp. 433–442, 2020.

[7] M. Shafiq, Z. Tian, A. K. Bashir et al., "Data mining and machine learning methods for sustainable smart cities traffic classification: a survey," *Sustainable Cities and Society*, vol. 60, 2020.

[8] Y. Han, C. Liu, S. Su, M. Zhu, Z. Zhang, and S. Zhang, "A proactive service model facilitating stream data fusion and correlation," *International Journal of Web Services Research*, vol. 14, no. 3, pp. 1–16, 2017.

[9] H. Ren, B. Xu, Y. Wang et al., "Time-series anomaly detection service at microsoft," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3009–3017, Anchorage, AK, USA, August 2019.

[10] L. Stojanovic, M. Dinic, N. Stojanovic et al., "Big-data-driven anomaly detection in industry (4.0): an approach and a case study," in *Proceedings of the 2016 IEEE International Conference on Big Data (Big Data)*, IEEE, Washington, DC, USA, December 2016.

[11] Z. Zhang, J. Yu, X. Li et al., "A data-driven service creation approach for effectively capturing events from multiple sensor streams," in *Proceedings of the 2019 IEEE International*

Conference on Web Services (ICWS), pp. 346–354, IEEE, Milan, Italy, July 2019.

[12] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[13] M. Braei and S. Wagner, "Anomaly detection in univariate time-series: a survey on the state-of-the-art," Preprint, 2020.

[14] J. Bu, Y. Liu, S. Zhang et al., "Rapid deployment of anomaly detection models for large number of emerging KPI streams," in *Proceedings of 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*, pp. 1–8, IEEE, Orlando, FL, USA, November 2018.

[15] Z. Yang, W. Ding, Z. Zhang, H. Li, M. Zhang, and C. Liu, "A service selection framework for anomaly detection in IoT stream data," in *Proceedings of 2020 International Conference on Service Science*, pp. 155–161, ICSS, Xining, China, August 2020.

[16] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time series feature extraction on basis of scalable hypothesis tests (tsfresh–A python package)," *Neurocomputing*, vol. 307, pp. 72–77, 2018.

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

[18] Yahoo! Webscope Dataset Ydata-Labeled-Time-Series-Anomalies-V10 2010, http://labs.yahoo.com/Academic_Relations.

[19] E. Ostertagova and O. Ostertag, *The Simple Exponential Smoothing Model*, Monash University, Melbourne, UK, 2011.

[20] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[21] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proceedings of ACM International Conference Proceeding Series*, pp. 4–11, New York, NY, USA, December 2014.

[22] P. Malhotra, A. Ramakrishnan, G. Anand et al., "LSTM-based encoder-decoder for multi-sensor anomaly detection," arXiv Prepr arXiv160700148, 2016.

[23] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial Databases with Noise," in *Proceedings of KDD-96*, Munchen, Germany, 1996.

[24] F. T. Liu, K. M. Ting, and Z. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 1, pp. 1–39.

[25] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof," *ACM SIGMOD Record*, vol. 29, no. 2, pp. 93–104, 2000.

[26] J. Ma and S. Perkins, "Time-series novelty detection using one-class support vector machines," in *Proceedings of the International Joint Conference on Neutral Networks*, Los Alamos, New Mexico, July 2003.

[27] J.-B. Kao and J.-R. Jiang, "Anomaly detection for univariate time series with statistics and deep learning," in *Proceedings of 2019 IEEE Eurasia Conference on IOT, Communication and Engineering*, pp. 404–407, ECICE), Yunlin, Taiwan, October 2019.

[28] A. Bagozi, D. Bianchini, V. De Antonellis, M. Garda, and A. Marini, "A relevance-based approach for big data exploration," *Future Generation Computer Systems*, vol. 101, pp. 51–69, 2019.

[29] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *Proceedings of Knowledge Discovery and Data Mining*, pp. 785–794, San Francisco, CF, USA, August 2016.

[30] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *Proceedings of Knowledge Discovery and Data Mining*, pp. 1939–1947, Sydney, UK, August 2015.

[31] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, 1998.

[32] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017.