MŰEGYETEM 1782

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Networked Systems and Services

Machine Learning-Based Anomaly Detection on Pre-processed and Decomposed Data

Scientific Students' Association Report

Author:

Dániel László Vajda

Advisor:

dr. Károly Farkas

2022

Contents

Kivonat

Napjainkban a hálózati eszközök, rendszerek és szolgáltatások folyamatos felügyelete lényegesebb, mint valaha. Ennek számos haszna lehet, mint például üzemzavar előrejelzése; leállások elkerülése; rendszerek teljesítményének monitorozása; továbbá rendszerek biztonságának felügyelete és az esetleges támadások észlelése.

A begyűjtött adatok feldolgozásának egyik ígéretes módja a hibás működésre utaló jelek pontos azonosítása és valós idejű jelzése, azaz az anomáliadetekció. Ez a dolgozat kifejezetten erre a feladatra összpontosít, és új megvilágításba kívánja helyezni az idősor alapú telemetria adatokon való anomáliadetekciót. Erre a célra egyre többen gépi tanuló algoritmusokat alkalmaznak, melyek képesek megérteni, csoportosítani és értékelni az infrastruktúra elemeinek működését leíró információkat, akár jelentős adathalmazok esetén is. Azonban ezzel kapcsolatban még számos feladat vár megoldásra, mint például nagyméretű adatok hatékony előfeldolgozása.

A korszerű, idősorokon alkalmazott anomáliadetekciót megvalósító eljárásokra vonatkozóan végzett korábbi irodalomkutatásaim során az ún. Long-Short Term Memory alapú, ReRe elnevezésű, Ming-Chang Lee és társai által kidolgozott algoritmust azonosítottam, mint egyike a jelenleg elérhető leghatékonyabb valós idejű anomáliadetekciós eljárásoknak. Korábbi munkám során kidolgoztam ennek a ReRe algoritmusnak egy továbbfejlesztett és hatékonyabb változatát, az Alter-Re^2 algoritmust, mely képes szórványos modelltanítással is magas pontosságot elérni kevés hibás pozitív jelzés mellett. Azonban az Alter-Re^2 egyik hiányossága, hogy periodikus adatsorokon nem működik hatékonyan. Így a jelen munka során eme probléma kiküszöbölése volt a célom különböző előfeldolgozó, illetve adattranszformációs eljárások segítségével.

Jelen kutatásaim kiindulópontjaként a Konstantin Dragomiretskiy és tsa. által kifejlesztett Variational Mode Decomposition (VMD) elnevezésű eljárást választottam, mely egy modern, matematikailag mélyen megalapozott adatdekompozíciós módszer. A VMD az eredeti adatsort kis sávszélességű függvényekre (módusokra) bontja szét, melyek összege jól közelíti az eredeti adatokat. Hipotézisem szerint ezen módusokat kivonva az eredeti jelből olyan maradvány adatsort kapunk, melyből az idősor periodikus jellegét eltávolítva az továbbra is magán hordozza az anomáliák jellemző jegyeit.

Ennek validálására részletes és mélyreható kísérleteket végeztem, melyben az eredeti idősort először skálázással átalakítottam, ezt követően a VMD algoritmussal transzformáltam, végül pedig az így kapott adatokon lefuttattam a korábban kidolgozott Alter-Re^2 eljárást. Kísérleteim eredményeként sikerült az Alter-Re^2 algoritmus teljesítményét három tudományosan megalapozott metrika szerint is átlagosan kétszeresre növeni, igazolva ezzel a módszerben rejlő jelentős potenciált.

Abstract

System state monitoring has become critical at multiple system layers, from physical network infrastructure to services. Performance monitoring plays a crucial role in malfunction detection, predicting and preventing system downtimes.

One of the most promising applications of monitoring is anomaly detection. Should the monitored system behave abnormally, unusual patterns in the data, i.e., anomalies, occur. Identifying such anomalous patterns is called anomaly detection, which is especially useful when done in real time. Machine learning can process and analyse infrastructure behaviour, even in large data volumes. The use of machine learning techniques in anomaly detection is emerging. However, many obstacles are yet to be tackled, such as the pre-processing or transformation of vast amounts of data.

In our previous state-of-the-art research, we found ReRe, a Long Short-Term Memory based machine learning algorithm by Ming-Chang Lee et al., to be one of the most promising approaches for real-time anomaly detection on network time-series data. In our previous work, we proposed an improved version of ReRe called Alter-Re^2. This algorithm offers high anomaly detection accuracy with a low number of false positives and retrains. However, it still has shortcomings, such as its poor performance on periodic datasets. Thus, this study focuses on pre-processing and data transformation approaches that facilitate extending Alter-Re^2 into the periodic data domain.

Towards this direction, we have identified a modern, theoretically well-grounded pre-processing approach named Variational Mode Decomposition (VMD) by Konstantin Dragomiretskiy et al. It separates the dataset into a set of functions with low spectral bandwidth, whose sum approximates the original data. We hypothesise that we can extract the periodic nature of the data while conserving the signs of anomalies by removing all of these mode functions from the original time series.

To test our hypothesis, we conducted experiments using scaling and our VMD-based approach, then running Alter-Re^2 on the transformed data. We managed to double the performance of Alter-Re^2 in three, theoretically well-grounded metrics, proving the potential of our approach.

Chapter 1

Introduction

Network system monitoring has become a crucial issue with the increase in complexity on all levels. While it is inevitable on the physical level, services and applications can also greatly benefit from performance monitoring. It can serve multiple purposes including malfunction detection, prediction and prevention of system outages, performance logging and intrusion detection.

The sheer number of interconnected devices in a network infrastructure makes it increasingly difficult to achieve reliable and robust infrastructure monitoring. Although challenging, it is undoubtedly necessary to continuously assess complex processes, uncovering details about the parts' influence on the system itself and its stability. The development of network telemetry as a concept is a major step in this direction. It allows automatic, fast and parallel streams of information to be received and stored continuously, containing time series data from a wide variety of sources and data types.

Even though solutions for reliable, scalable real-time monitoring are emerging, the continual analysis of these large amounts of data points preferably also in real-time is under active research today, and poses great challenges regarding time- and resource-constraints.

An increasingly important use case for processing telemetry data is anomaly detection, whereby the aim is to identify data points and patterns that reflect erroneous behaviour. This can be the result of network device or sensor malfunction, but might come in the form of a drop in temperature or a change in traffic. Therefore, the larger the number of sources that can be analysed by anomaly detection, the more use cases it can support.

In the last decade, machine learning techniques have increasingly become the approach the majority employ for the aim of anomaly detection, as it is capable of analysing and learning complex patterns present even in a large dataset. There are still, however, challenges to be overcome.

Through our research into the state of the art, we found the algorithm ReRe [1] by Lee et al. to be one of the most promising real-time anomaly detection engines. It, like a significant amount of other approaches, utilises a Long Short-Term Memory (LSTM) neural network for prediction. Earlier we published an improved version of ReRe, named Alter-Re^2 [2], which achieved high precision and a small number of false positives and LSTM model retrains.

Through our research into the operation of Alter-Re^2, however, we identified that it performed noticeably better on certain types of data than others. Such new findings, along with other improvements are presented in a new article by our research team under publication [3]. Therein, we classify datasets based on their periodicity and spikiness, and present results showing the drop in performance a periodic or spiked dataset brings to Alter-Re^2.

In this study, our aim therefore is to extend Alter-Re^2 operation to spiked, but especially periodic datasets to broaden its number of use cases and deployment scenarios. To facilitate this goal, we have identified a state-of-the-art, mathematically well-grounded pre-processing approach, named Variational Mode Decomposition (VMD) [4] by Dragomiretskiy and Zosso. VMD can decompose any dataset into a set of functions with low spectral bandwidth, called modes, whose sum approximates the original dataset. Through our analysis of this decomposition, we hypothesised that by subtracting these functions from the original data, the residue remaining will still contain almost all signs of anomalies. At the same time, given that modes contain regular, periodic data, periodicity would have been almost entirely eradicated from the data, approximately none remaining in the residue.

Combining scaling with this approach, we evaluated different parameter settings, feeding this pre-processed data to Alter-Re^2, and comparing it with Alter-Re^2 run on the original data with no transformation. We found that Alter-Re^2 performed at least twice as good with our pre-processing approach as it did without it, proving the potential of our method. In our experiments, we used the well-established metrics of Precision, Recall and F-score to ensure reliable results.

The rest of this study is organised as follows. Section 2 outlines related work in the field of mode decomposition algorithms, whilst also discussing anomaly detection engines Alter-Re^2 is based on. Section 3 describes three algorithms we utilise in our approach, namely MinMax Scaler, VMD and Alter-Re^2. Afterwards, we discuss the limitations of Alter-Re^2 to motivate our new approach. In Section 4, we discuss the details of our pre-processing approach, describing its interoperability with Alter-Re^2. Section 5 presents our experimental setup and our results when assessing the improvement our pre-processing method brings to Alter-Re^2. We also discuss the implications of the results presented, and future areas of research that unfold from that. Lastly, in Section 6, we make our final comments to conclude the study.

Chapter 2

Related work

In this chapter, we discuss state-of-the-art data mode decomposition methods. We also review the collection of modern anomaly detection algorithms which our own, earlier developed approach is based on.

2.1 Mode decomposition algorithms

Empirical Mode Decomposition (EMD) is an algorithm developed by Huang et al. [5] that can decompose any dataset into a set of Intrinsic Mode Functions (IMFs). Their paper, published in 1998, also includes an extensive analysis of non-stationary time series based on this decomposition and the Hilbert spectrum. Huang et al. define IMFs using two simple criteria: their number of extrema can only differ at most by one from their number of zero crossings; and their mean envelope must be zero. EMD uses a recursive sifting method to extract IMFs from the original data. It starts from the highest frequencies, and iteratively subtracts modes as long as it is possible, until only a relatively unimportant final residue if left. While EMD is simple to implement, and is the foundation for further research, it suffers from issues with regard to noise tolerance and accurate separation of modes (mode mixing). As a result of its simplicity, it is also less grounded in mathematical theory.

Ensemble Empirical Mode Decomposition (EEMD) is an improved version of EMD by Wu and Huang [6]. EEMD employs a noise-assisted data analysis (NADA) approach, whereby an ensemble of signals are created from the original signal by introducing various degrees of white noise. Modes then are decomposed from each signal from the ensemble, and final IMFs are achieved by averaging these modes. Their goal is to solve the mode-mixing problem, while preserving physical uniqueness of certain decomposed signals. Although successful in their aim, EEMD tends to leave residual noise in the final modes, and might produce a different number of modes for the same original input dataset depending on realisation.

Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) by Torres et al. [7] attempts solve these exact issues. This is done by adding a particular noise at each stage of decomposition, and computing a unique residue to obtain each mode. CEEMDAN also achieves better spectral separation of modes, and a reduction in resource demands through a smaller number of iterations compared to EEMD. Its recursive sifting method, however, is preserved.

Empirical Wavelet Transform (EWT), developed by Gilles [8] has the same aim of mode decomposition, i.e. extracting low-bandwidth IMFs from the original data with a compact Fourier support, like EMD and its improvements. EWT, however, bases its structure in significantly more complex mathematical theory than the previous approaches outlined. While traditional wavelet transform algorithms make use of predefined filter banks, EWT calculates the support of its filters in a fully adaptive fashion, based on the spectrum of the original input signal. EWT requires the a-priori selection of the number of modes N, then separates the smoothed spectrum of the original signal into N regions, drawing boundaries at midpoints between the $N - 1$ spectral maxima. EWT then applies these limits to its filter bank, and extracts the N modes from the signal.

Variational Mode Decomposition (VMD) by Dragomiretskiy and Zosso [4] supersedes EMD and improvements with its deep-rooted mathematical foundation and EWT by less strict boundaries in its filter bank. Published in 2014, it is an excellent mode decomposition approach with robust sampling and noise tolerance. VMD's adaptive calculation of modes' spectral bands replaces the recursive way EMD and its improvements determine IMFs, and arrives at its decision by concurrent iterations on all modes. Similarly to EWT, VMD suffers from the necessity to set the number of modes K a-priori. Nonetheless, we selected VMD for our pre-processing approach, and a more detailed description of its operation can be found in Section 3.2.

2.2 Anomaly detection algorithms

In this section, we refrain from an extensive look on all modern anomaly detection algorithms, as we believe it would not be in line with the focus of this study. Instead, we list anomaly detectors that are direct predecessors to our chosen approach, Alter-Re^2 [2], developed by our research team. For an extensive review of state-of-the-art anomaly detection engines, we redirect the reader to our article on Alter-Re^2 [2] or to another article by our research team currently accepted for publication [3] with updated references.

All algorithms listed below utilise a Long Short-Term Memory (LSTM) neural network for time series prediction. LSTMs, developed by Hochreiter and Schmidhuber [9] in 1997, are a type of Recurrent Neural Networks well-suited for this task, and are widely used throughout scientific research into anomaly detection.

The Greenhouse algorithm [10] by Lee et al. combines the above-mentioned LSTM with data management techniques for anomaly prediction over high volumes of time series. Although it still requires labelled data for its training, Greenhouse only needs normal samples for that purpose – an approach widely referred to as 'zero positive' or semi-supervised learning – and can be trained on a relatively small dataset. The LSTM model, based on previous data as input, predicts the next few timesteps. The actual data point is then compared to the predicted one to decide whether it constitutes an anomaly. This is called the 'look-back, predict-forward' approach, and all algorithms listed below utilise the same method.

RePAD [11] by Lee et al. is a direct improvement of Greenhouse that alleviates the need for labelled training data, and is capable of operating in real time completely unsupervised, while also being able to adapt to changing patterns in the data indefinitely.

ReRe [1] by the same authors further upgrades RePAD by introducing another detector, which operates only on normal (non-anomalous) data. ReRe only signals an anomaly if both detectors do so. This is aimed at reducing false positive detection signals.

Alter-Re^2 [2], developed by our research team is an improved version of ReRe, that reduces the algorithm's resource demands, while significantly improving its performance by introducing a sliding window and an ageing mechanism. This is the algorithm we chose to improve even further. More details on Alter-Re^2 can be found in Section 3.3.

Chapter 3

Background and motivation

In this chapter, we discuss the operation and parameters of three algorithms necessary to comprehend our approach detailed in Chapter 4, namely MinMax Scaler, Variational Mode Decomposition and Alter-Re^2. The fourth section is devoted to analysing limitations of Alter-Re^2 to motivate our improvements.

3.1 Scaling

Firstly, for the purpose of normalising our dataset, we chose the robust, simple and fast algorithm MinMax Scaler created by Pedregosa et al. [12].

Patro and Sahu in their article [13] describe its operation with Equation (3.1):

$$\mathbf{x}_{MinMax} = \frac{\mathbf{x} - min(\mathbf{x})}{max(\mathbf{x}) - min(\mathbf{x})} \cdot (\mathrm{MAX} - \mathrm{MIN}) + \mathrm{MIN}, \tag{3.1}$$

where

- $\mathbf{x}$ is the original dataset;
- $\mathbf{x}_{MinMax}$ is the MinMax scaled data;
- $min(\mathbf{x})$ is the minimum of the original dataset;
- $max(\mathbf{x})$ is the maximum of the original dataset;
- $[\mathrm{MIN}, \mathrm{MAX}]$ is the desired range of the scaled data.

As written in Equation (3.1), there are two input parameters for the scaler, MIN and MAX. These specify the desired value range of the output data.

MinMax Scaler data transformation can be described in three steps: firstly, the dataset is shifted vertically to have its minimum at zero; secondly, all values are divided by the distance between the maximum and minimum achieving a data range of $[0, 1]$; lastly, the data is scaled up and shifted to satisfy user input and achieve the data range of $[\mathrm{MIN}, \mathrm{MAX}]$.

3.2 Variational Mode Decomposition

Variational Mode Decomposition (VMD) is a non-recursive algorithm designed to concurrently extract modes from a dataset. Dragomiretskiy and Zosso [4] define Intrinsic Mode Functions (IMFs or modes in short) as AM-FM modulated signals in the form of:

$$u_k(t) = A_k(t) \cdot \cos \phi_k(t), \tag{3.2}$$

where

- $u_k(t)$ is the kth mode function;
- $A_k(t)$ is the envelope of the kth mode;
- $\phi_k(t)$ is the phase of the kth mode.

If the following three conditions are fulfilled for all values of t, the function described in Equation (3.2) is considered an IMF:

- $\omega_k(t) = \phi'_k(t) \geq 0$, where $\omega_k(t)$ is the instantaneous frequency of the kth mode;
- $A_k(t) \geq 0$;
- both $A_k(t)$ and $\omega_k(t)$ vary much slower than $\phi_k(t)$.

This definition ensures all modes have a limited bandwidth, which is the foundation upon which Dragomiretskiy and Zosso base their mode separation algorithm, VMD. For more discussion on the importance of IMF definition, the behaviour of such functions and bandwidth estimation, we refer the reader to [4].

Nonetheless, VMD decomposes the input dataset into a predefined number (K) of modes. The set of modes is denoted by $\{\mathbf{u_k}\}$, and the set of their respective center frequencies are denoted by $\{\omega_k\}$.

The core of VMD solves a constrained variational optimization problem employing the alternate direction method of multipliers (ADMM) to produce optimal selection of IMFs ($\{\omega_k, \mathbf{u_k}\}$ pairs). The algorithm goes through the following steps:

1. apply the Hilbert transform [14] to the original dataset, then produce a so-called single-side band analytic signal;
2. shift each mode function to baseband frequency through complex harmonic mixing;
3. calculate the bandwidth of each mode using the squared L^2 norm of the gradient.

This constrained problem thus constructed is shown in Equation (3.3):

$$\begin{gathered} \min_{\{\mathbf{u_k}\},\{\omega_k\}} \left\{ \sum_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] \cdot e^{-j\omega_k t} \right\|_2^2 \right\} \\ \text{s.t.} \sum_{k=1}^{K} u_k(t) = v(t), \end{gathered} \tag{3.3}$$

where

- $\left(\delta(t)+\frac{j}{\pi t}\right)*u_k(t)$ is the analytic signal of $u_k(t)$ achieved using the Hilbert transform;
- $x \cdot e^{-j\omega_k t}$ is the baseband-shifted signal of x, where ω_k is the centre frequency of the kth mode;
- $\partial_t x$ is the gradient of x in time;
- $\|x\|_2^2$ is the L^2 norm of x;
- $v(t)$ is the original input signal.

To transform this into an unconstrained problem, Dragomiretskiy and Zosso introduced a quadratic penalty term along with Lagrangian multipliers. Equation (3.4) shows the augmented Lagrangian $\mathcal{L}$:

$$\begin{aligned}\mathcal{L}\left(\{\mathbf{u_k}\},\{\omega_k\},\lambda\right) := \alpha \cdot \sum_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t)+\frac{j}{\pi t}\right) * u_k(t) \right] \cdot e^{-j\omega_k t} \right\|_2^2 + \\ \left\| v(t) - \sum_{k=1}^{K} u_k(t) \right\|_2^2 + \left\langle \lambda(t), v(t) - \sum_{k=1}^{K} u_k(t) \right\rangle,\end{aligned} \tag{3.4}$$

where

- $\mathcal{L}$ is the augmented Lagrangian;
- λ is the Lagrangian multiplier;
- α is the bandwidth coefficient.

Solving Equation 3.4 using ADMM, we get the method through which to update all K modes in one iteration $n \to n+1$. This is presented in Equation (3.5) in the frequency domain:

$$\hat{u}_k^{n+1}(\omega) = \frac{\hat{f}(\omega) - \sum_{i<k} \hat{u}_i^{n+1}(\omega) - \sum_{i>k} \hat{u}_i^{n}(\omega) + \frac{\hat{\lambda}^n(\omega)}{2}}{1 + 2\alpha \cdot (\omega - \omega_k^n)^2}. \tag{3.5}$$

The iteration update presented in Equation (3.5) is an application of simple Wiener filtering. Finally, Equation (3.6) shows one iteration of updating central frequencies:

$$\omega_k^{n+1} = \frac{\int_0^\infty \omega \left|\hat{u}_k(\omega)\right|^2 d\omega}{\int_0^\infty \left|\hat{u}_k(\omega)\right|^2 d\omega}. \tag{3.6}$$

To conclude, due to the necessary introduction of the quadratic penalty and the Lagrangian multiplier to make the problem unconstrained, the sum of modes deconstructed by VMD no longer equals the original data, only approximates it to the desired degree: $\sum_{k=1}^{K} u_k(t) \approx v(t)$.

In order to use VMD, the following parameters need to be set in advance:

- α: the balancing parameter of the data-fidelity constraint, it has a direct influence on the bandwidth of modes;
- τ: time-step of the dual ascent, determines how Lagrangian multipliers are used;

- K: the number of modes;
- DC: whether to fix the first mode to zero frequency;
- init: one of three initialization scenarios for centre frequencies ($\{\omega_k\}$):
    1. all centre frequencies start at zero;
    2. all centre frequencies are uniformly distributed;
    3. all centre frequencies are initialised randomly;
- ϵ: tolerance of convergence.

3.3 Alter-Re^2

Alter-Re^2 is an anomaly detection algorithm developed by our research team, published in 2021 [2]. It is an improved version of ReRe [1] which in turn is based on RePAD [11], both of which were developed by Lee et al.

At the core of Alter-Re^2 is a so-called 'look-back, predict-forward' approach, whereby a Long Short-Term Memory (LSTM) neural network predicts the upcoming data point based on previous values. Once the new data point arrives at the detector, it is compared with the prediction to make a decision on whether it constitutes an anomaly or a pattern change in the data.

LSTMs are a type of Recurrent Neural Networks that were chosen as the prediction model due to their excellent performance on time series datasets. Similarly to RePAD [11] and ReRe [1], the LSTM model used is lightweight with a small number of neurons, epochs, input data and learning rate to facilitate real-time operation.

The ability of Alter-Re^2 to adapt its detection threshold and neural network to changing patterns in the data is inherited from ReRe and makes it well suited for long-term online deployment, which is further aided by it being ready for detection soon after startup. Another advantage is that the LSTM model is trained in an unsupervised fashion, therefore not requiring any labelled data throughout its operation.

Alter-Re^2 introduced two key improvements to ReRe; a sliding window and an ageing mechanism. This way, it achieved phasing out old data points that had a negative impact on the speed and precision of anomaly detection, while also limiting resource requirements. The size of the sliding window is denoted by WS, while ageing is controlled by the age power parameter, AP.

After an initial training period for which the LSTM model takes n_{epochs} number of epochs per timestep, Alter-Re^2 repeats the same actions every timestep. These are depicted in Figure 3.1. The first step is utilizing the LSTM model, having n_{neurons} hidden neurons, to predict the data point expected to arrive in the current timestep, $\widehat{v_t}$, based on the previous b data points. Therefore, b is the so-called look-back parameter. We must note that Alter-Re^2 always predicts one single data point ahead, just like its predecessors.

Afterwards, Alter-Re^2 calculates the beginning timestep for the sliding window using Equation (3.7). The method outlined effectively fixes the beginning to timestep b, from where the first values are available, until there is more than enough available data points based on the size of the sliding window.

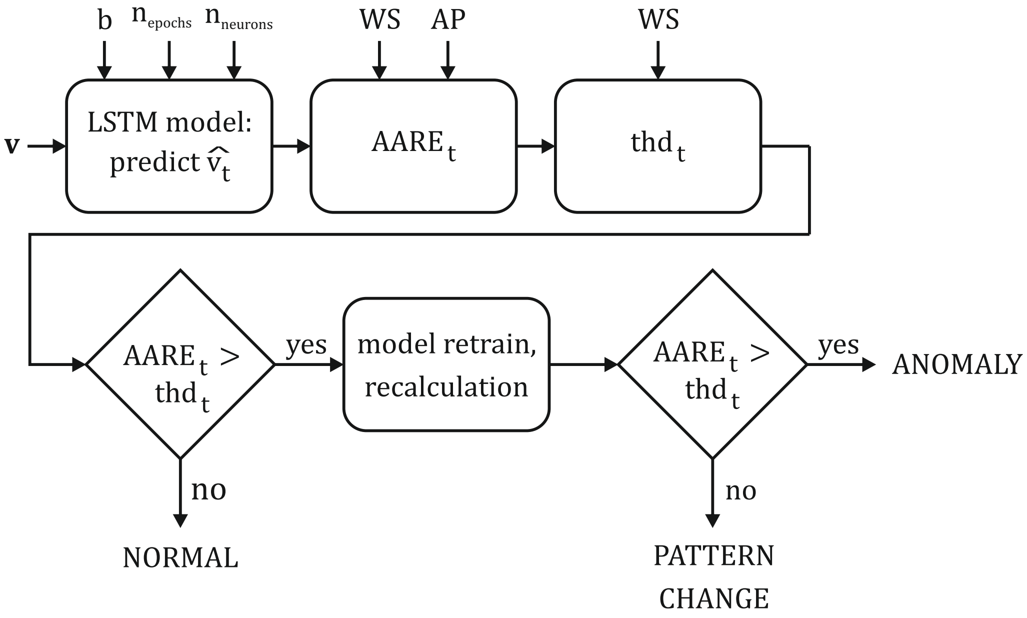


Figure 3.1: One step of Alter-Re^2 operation

$$W = \begin{cases} t - WS + 1 & \text{if } t > b + WS - 1 \\ b & \text{otherwise} \end{cases}, \tag{3.7}$$

where

- W is the beginning timetep of the sliding window;
- t is the current timestep;
- WS is the size of the sliding window (parameter);
- b is the look-back parameter.

The other improvement, ageing, is introduced via the ageing coefficient, C_y, calculated in Equation (3.8). AP controls the aggressivness of ageing; the higher it is, the less old data points are taken into account for error calculation.

$$C_y = \left(\frac{y - W}{t - W}\right)^{AP}, \tag{3.8}$$

where

- C_y is the ageing coefficient at timestep y;
- AP is the age power parameter.

The next step for Alter-Re^2 is to calculate the Average Absolute Relative Error of prediction using Equation (3.9). We point out that summation is only performed within the sliding window, and all absolute relative error terms are multiplied by C_y to achieve ageing of data points that arrived earlier.

$$AARE_t = \frac{1}{t - W + 1} \cdot \sum_{y=W}^{t} C_y \cdot \frac{|v_y - \widehat{v_y}|}{v_y}, \tag{3.9}$$

where

- $AARE_t$ is the Average Absolute Relative Error at current timestep t;
- v_y is the data point of the original input dataset at timestep y;
- $\widehat{v_y}$ is the predicted data point at timestep y.

After calculating $AARE_t$, Alter-Re^2 evaluates the average of all $AARE$ values (Equation (3.10)), along with their standard deviation (Equation (3.11)). Both are performed within the sliding window. Detection threshold is calculated utilising the Three Sigma Rule shown in Equation (3.12).

$$\mu_{AARE,t} = \frac{1}{t-W+1} \cdot \sum_{y=W}^{t} AARE_y, \tag{3.10}$$

where

- $\mu_{AARE,t}$ is the average of $AARE$ values in the sliding window at timestep t.

$$\sigma_{AARE,t} = \sqrt{\frac{1}{t-W+1} \cdot \sum_{y=W}^{t} \left(AARE_y - \mu_{AARE,t}\right)^2}, \tag{3.11}$$

where

- $\sigma_{AARE,t}$ is the standard deviation of $AARE$ values in the sliding window at timestep t.

$$thd_t = \mu_{AARE,t} + 3 \cdot \sigma_{AARE,t}, \tag{3.12}$$

where

- thd_t is the detection threshold at timestep t.

As seen in Figure 3.1, after these calculations, Alter-Re^2 compares the current error value $AARE_t$ to the current threshold value thd_t. As long as the error is below threshold, current timestep t is regarded as normal. If not, Alter-Re^2 retrains its LSTM model and recalculates both $AARE_t$ and thd_t. If the error is now below threshold, that means the new model trained on recent data managed to accurately predict the current data point, while the old one could not. Alter-Re^2 decides for a pattern change, and keeps the new LSTM model. If the error is still above threshold, Alter-Re^2 notifies the user of an anomaly, and continues using the old LSTM model.

These operation steps summarised in Figure 3.1 are repeated in every timestep. Importantly, Alter-Re^2 utilises two detectors that it inherited from ReRe [1], thus having two separate LSTM models for prediction with the second detector only taking in normal (non-anomalous) timesteps for $AARE_t$ calculation.

3.4 Limitations of Alter-Re^2

Our original configuration using Alter-Re^2 detailed in the previous section is summarized in Figure 3.2 along with its hyperparameters required to be set.

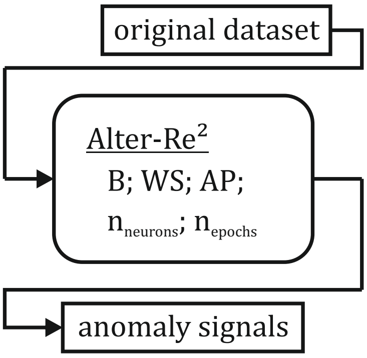


Figure 3.2: Original Alter-Re^2 configuration with the corresponding parameters

In a new article of our research team under publication [3], we assess Alter-Re^2 operation on different types of data. Based on experimental results outlined in the paper, we can conclude that Alter-Re^2 performs best on aperiodic datasets with a constant rolling average where anomalies present as spikes or shifts in this average. For datasets with periodic data patterns, especially where periodic spikes are part of normal operation, Alter-Re^2 regualarly achieves worse performance.

This is in part due to regular spikes raising the detection threshold so high that, even with aggressive ageing, error values do not rise above the threshold function, Alter-Re^2 thus missing anomalies entirely.

Furthermore, the aim of real-time operation limits the complexity of the LSTM model in use. More specifically, while increasing the look-back parameter b would definitely allow Alter-Re^2 to comprehend periodic data with a larger period (still smaller than b, naturally), it would dramatically increase time and resource demands. In the case where hundreds or thousands of data points are required to accurately predict the following one, model training times would be unacceptable even on state-of-the-art hardware.

The same is true for the number of neurons in the hidden layer, where a higher n_{neurons} value would allow for a more complex neural network capable of learning more nuanced data patterns, but it would also result in such high infrastructural demands that they would prevent any applicability in actuality.

Having identified the type of data Alter-Re^2 operates best on, while having concluded that there are inherent limitations in the design of the algorithm to extend its applicability to periodic, especially spiked data pattern types, we decided to focus on the input dataset instead. Our goal thus became to transform datasets into a form optimal for Alter-Re^2, while paying special attention to preserving anomalies from the data in the process.

Chapter 4

Proposed pre-processing procedure

In this chapter, we outline our approach to improve Alter-Re^2 performance via the scaling and mode decomposition of the input data. The structure of our approach is outlined in Figure 4.1.

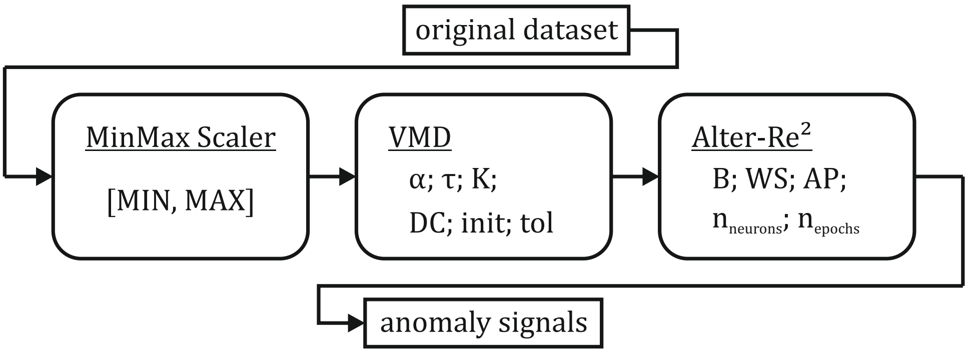


Figure 4.1: Operation flow of our proposed procedure

The first section discusses removing modes from the original data produced by VMD to alleviate the issues regarding periodic patterns, while the second section explains how we combine this method with scaling to form our pre-processing approach for Alter-Re^2.

4.1 Removing modes from data

As discussed in Section 3.2, VMD is capable of separating an input time series into intrinsic mode functions whose sum approximates the original dataset.

These modes capture patterns of regularity, and often of periodicity. Our hypothesis was that signs of anomalies in the original data would not transfer to the modes given their nature, as they are defined as low-bandwidth AM-FM signals. We managed to confirm this assumption in our experiments in Section 5.

Given that the sum of modes only approximates the original dataset, there is a slight, but important difference between the two. This data remaining after removing all modes depends on parameter values of VMD. Nonetheless, we can define the so-called residue data to be equal to the original data minus the sum of all modes. This is shown in Equation(4.1):

$$\mathbf{r} = \mathbf{v} - \sum_{k=1}^{K} \mathbf{u_k}, \tag{4.1}$$

where

- $\mathbf{r}$ is the residue dataset;
- $\mathbf{v}$ is the original dataset;
- $\mathbf{u_k}$ is the kth mode;
- K is the number of modes (parameter).

Stated that VMD modes only contain the minority of anomalous behaviour, almost all anomalies present in the original data $\mathbf{v}$ will be preserved in the residue $\mathbf{r}$, while also having removed regular, periodic patterns of normal behaviour.

This way, Alter-Re^2 performance can be improved by running it only on $\mathbf{r}$, having removed all decomposed modes from the original data.

4.2 Components

In this section, we discuss how we combine the components discussed earlier to constitute our pre-processing and anomaly detection pipeline.

To improve our original setup, we combined MinMax scaling (see Section 3.1) with VMD (Section 3.2). Thus, the input data for Alter-Re^2 is not the original dataset itself, but it is the residue time series $\mathbf{r}$ achieved by the method in Equation (4.1).

The motivation for scaling is the property of neural networks to operate best in a predefined data range. This is usually set either as $[0, 1]$ or $[-1, 1]$, depending on the type of neural network employed.

On the other hand, the aim of utilising VMD is to remove periodicity to the limit possible from the original dataset, and supply Alter-Re^2 with data it is better equipped to detect anomalies on. Contrast Figure 3.2 displaying our original setup with Figure 4.1 that shows this pre-processing approach combining MinMax scaling with VMD. The set of parameters to be set is also displayed.

The order of components (i.e. MinMax scaling before mode decomposition and subtraction) was chosen to aid VMD operation, as scaling in an appropriate way can automatically remove the DC component from the original dataset by shifting the data average to zero. This way, VMD is not required to waste a mode on representing the DC offset value. For more information on selection of required hyperparameters, refer to Section 5.1.2.

Chapter 5

Experiments

In this chapter we lay out our experiments conducted to evaluate the performance improvement our pre-processing procedure brings to the original Alter-Re^2 algorithm in six different test cases. In the first one we asses Alter-Re^2 without pre-processing as a baseline, while in the following five, we use different parameter settings for our mode decomposition approach.

5.1 Preliminaries

In this section, we lay out the specifics of our experimentation, including implementation, parameter settings, the datasets used and our evaluation metrics.

5.1.1 Implementation

All our code was implemented in the Python 3 language. For scaling, we used a popular machine learning and pre-processing library called Scikit-learn [12], more details on the parameters and use are included in the package documentation for their MinMaxScaler [15].

As for VMD, Carvalho et al. [16] have made their algorithm implementation publicly available, and also included their code in the standard Python 3 library called ‘vmdpy’ via PyPI [17]. We greatly appreciate their efforts with the excellent package.

Finally, Alter-Re^2 has been implemented by our research team discussed in [2]. Therefore, the Python 3 code was directly accessible. We are releasing our baseline implementation of Alter-Re^2 and its successor in a new article of our research group [3].

5.1.2 Parameter settings

As discussed in the introduction of Chapter 5, we conducted six experiments on all datasets; one baseline with the original Alter-Re^2 algorithm, and five experimental test cases using our pre-processing procedure with different parameter settings.

For all five parameter settings, we used the same scaling values detailed in Table 5.1. For the baseline original experiment, scaling was not used. It is well known throughout state-of-the-art research that neural networks perform best on datasets where all data points have an absolute value smaller than one.

Table 5.1: MinMax Scaler parameter settings for all test cases where scaling is used

Parameter	Value
MIN	-1.0
MAX	1.0

Similarly, VMD was not present in the baseline experiment either. For the five other test cases, we detailed our parameter settings in Table 5.2. Note that we decided to adjust only the α (mode bandwidth control) and K (number of modes) parameters, while setting all others to a fixed value.

Table 5.2: VMD parameter settings for each test case

	Setting				
Parameter	No. 1	No. 2	No. 3	No. 4	No. 5
α	500	500	500	100	100
τ			0.0		
K	12	8	4	12	8
DC			False		
init			1		
ϵ			10^{-7}		

Before the experiments, we had conducted heuristic tests to determine the range of VMD parameters acceptable in our use case. We found that only α and K directly influenced the results. We chose α to be acceptable below 1000, resulting in a relatively sizeable accepted mode bandwidth. We found K to be most effective around the value 10.

As for the other parameters, DC (whether to fix the first mode to zero frequency) is made irrelevant by our scaling approach, while the τ parameter (Lagrangian coefficient; approach to noisy data), initialization scenario, and tolerance settings (ϵ) also did not make a significant difference as to the algorithm performance, and were selected according to Carvalho et al. [16].

Lastly, Alter-Re^2 hyperparameter values were determined based on our latest research and selected according to the guidelines in [2]. These values (see Table 5.3) see are valid for all six test cases.

Table 5.3: Alter-Re^2 parameter settings for all tests

Parameter	Value
b	30
WS	800
AP	2.5
n_{neurons}	30
n_{epochs}	30

5.1.3 Datasets

We selected the Numenta Anomaly Benchmark (NAB) [18] to be the source of data for our experiments. The NAB consists of 58 datasets, which were curated by the team at Numenta Inc. [19]. Every dataset is a one dimensional expert-labelled time series data from various sources containing different types of anomalies. Telemetry sources include AWS server metrics, online advertisement clicking rates, temperature, CPU utilisation, keystroke and traffic information, along with some network-related and simulated sources.

Table 5.4 shows a few details of these datasets. On average they contain about 6300 timestamped values, and 2 labelled anomalies. There are a few datasets that were intentionally selected by Numenta to contain no anomalies, thus helping evaluate any algorithm's performance under normal (non-anomalous) circumstances.

Table 5.4: NAB details considering all 58 datasets

Metric	Dataset length	No. of anomalies
Average	6303	2.07
Minimum	1127	0
Maximum	22 695	5
Standard deviation	5525	1.30

We conducted six test cases (one baseline and five new ones) for each of the 58 datasets, thus 348 experiments in total, which, we believe, proves the wide applicability of our results detailed in Section 5.2.

5.1.4 Anomaly detection metrics

There is a wide selection of metrics used by researchers to evaluate anomaly detection algorithms. Of these, we selected the traditional, well-established metrics of Precision, Recall and F-score. These values all fall in the range $[0.0, 1.0]$ and are calculated by Equations (5.1), (5.2) and (5.3) respectively:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{5.1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{5.2}$$

$$\text{F-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{5.3}$$

where

- TP is the number of true positive detections;
- FP is the number of false positive detections;
- FN is the number of false negative detections.

Determining the values in the so-called confusion matrix (true / false positives and true / false negatives), however, is not as straightforward for anomaly detectors as it might be for other types of classification algorithms and is not done uniformly throughout the field's literature.

One of the key reasons for this is the varying tolerance to the delay between an anomaly detection signal and the timestep the anomaly in fact happened. This might vary case by case. Another factor is the handling of detection signals present for multiple timesteps.

Based on our research of the state of the art, we selected the approach of Lavin and Ahmad [20] as our basis on the issue. They designate so-called anomaly windows with length K_a around the ground truth anomaly labels. K_a is calculated by taking 10% of the dataset length and dividing this value by the number of ground truth anomalies present in the dataset. This value is then rounded up to the nearest integer.

Thus, we accept anomaly detections to be correct in the range $[T_a - K_a, T_a + K_a]$, where T_a is the labelled anomaly timestep. Only the first signal is counted as a true positive within the anomaly window. If there are no signals within it at all, the number of false negatives is increased by one. Outside the windows, every signal is regarded as false positive, while the remaining timesteps count as true negatives. We normalise these metrics outside the anomaly window by the number K_a to avoid the imbalance that arises from only counting one TP or FN value per anomaly window.

We also note that we only take into account the first timestep of each anomaly detection signal. Consequently, longer duration signals still only count as one timestep.

To sum up, we determine the length K_a; designate anomaly windows; measure the confusion matrix; then calculate the metrics Precision, Recall and F-score. We use these to compare our test cases for all 58 NAB datasets.

5.2 Evaluation of Alter-Re^2 with and without our pre-processing procedure

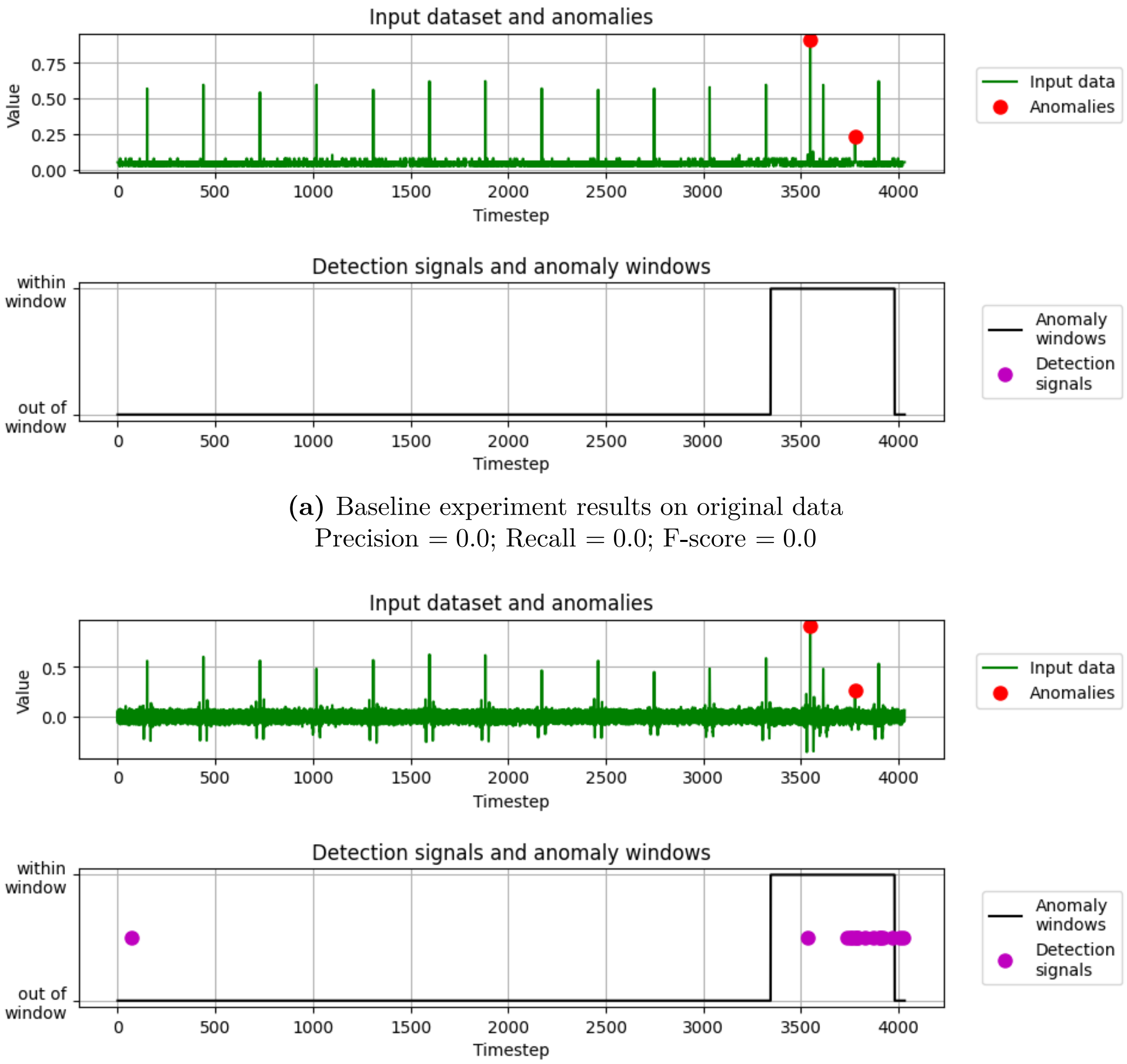
In this section, we present the results of our experiments. We begin with two datasets as examples, through which we explain our findings in detail. Afterwards, we present the comparison that resulted from using all 58 NAB datasets.

As discussed in [2] and [3], the ReRe family of algorithms in their current form lack the capacity to deal with complex periodic datasets. This is partially due to the lightweight LSTM model, that would need an increased number of neurons and input data. This would, however, prevent real-time operation through drastically increased resource demands.

Therefore, our aim was to remove normal periodicity through pre-processing, while maintaining, if not highlighting anomalies. Figure 5.1 shows experiment results on the dataset '`ec2_cpu_utilization_24ae8d`'.

For both subfigures, the top axis shows the input data fed to Alter-Re^2 in green, while labelled ground truth anomalies are denoted with a red circle. The bottom axis shows the position of anomaly windows around the labelled anomaly in black, while detection signals are drawn in magenta ink. In the case of this dataset, the two anomaly windows would overlap given their calculated length. In such scenarios, we set the end of the first window and the beginning of the next one at the halfway point between the anomalies.

The original NAB dataset '`ec2_cpu_utilization_24ae8d`' is shown in Figure 5.1a in the top axis. The time series consists of highly periodic spikes with similar amplitudes surrounded by noise. As a consequence, anomalies present as out-of-period and/or different



(a) Baseline experiment results on original data
Precision = 0.0; Recall = 0.0; F-score = 0.0

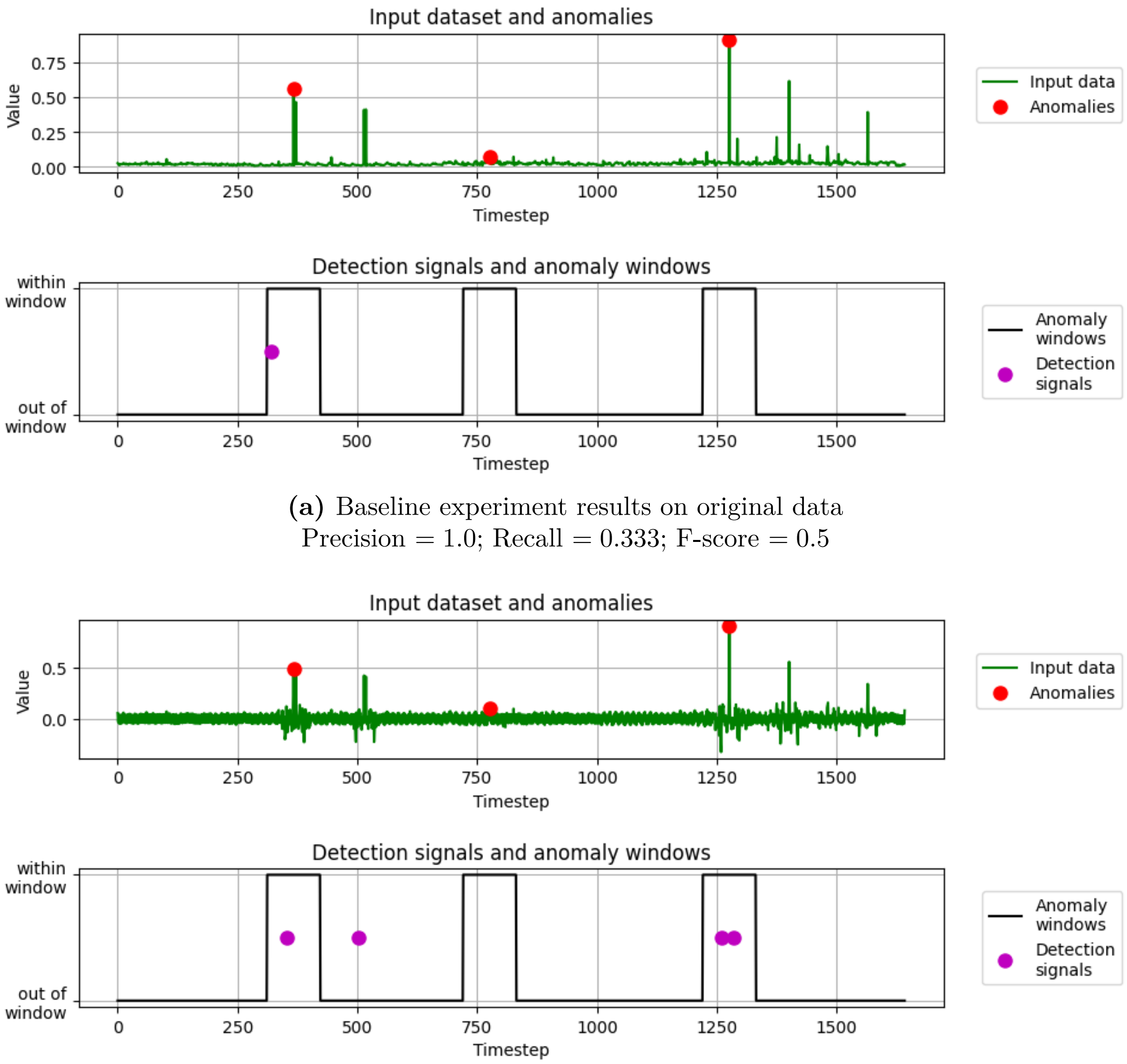
(b) Experiment results using our pre-processing procedure in Setting No. 4
Precision = 0.996; Recall = 1.0; F-score = 0.998

Figure 5.1: Experiment results for the NAB dataset
'`ec2_cpu_utilization_24ae8d`'

amplitude spikes, as seen in the figure. Unfortunately, the original Alter-Re^2 was not able to detect either, in fact making no anomaly signals at all (shown in the bottom axis). The most likely reason for this is that the spikes raise the average *AARE* value significantly, resulting in a high threshold function. And since they are only present for a few timesteps, the *AARE* error function has no chance of reaching it even with aggressive ageing. All three of our chosen metrics thus evaluate to zero.

The effects of our pre-processing procedure are shown in Figure 5.1b. The top axis displays the scaled and decomposed dataset, while the bottom axis shows Alter-Re^2 detections. Apart from a single false positive signal at the beginning of the dataset (around timestep 70), and a few at the very end, no erroneous signals are raised. Moreover, both anomalies are detected quickly after they occur, resulting in 100% Recall. Precision is also almost maximal with the value of 0.996.

In general, our experience with similar periodic and/or spiked types of data shows that Alter-Re^2 greatly benefits from our scaling and decomposition approach.



(a) Baseline experiment results on original data
Precision = 1.0; Recall = 0.333; F-score = 0.5

(b) Experiment results using our pre-processing procedure in Setting No. 4
Precision = 0.996; Recall = 0.667; F-score = 0.799

Figure 5.2: Experiment results for the NAB dataset ‘`exchange-4_cpc_results`’

The results for our second example dataset, ‘`exchange-4_cpc_results`’ are presented in Figure 5.2. Similarly to the previous example, the top subfigure, Figure 5.2a shows details without pre-processing. The original dataset is a collection of online advertisement clicking rates, which might explain the unexpected patterns of behaviour present in it. There are three labelled anomalies, although the second one is significantly smaller in magnitude than the other two. We conclude this second anomaly must therefore denote the absence of another peak, which is one of the anomaly types most challenging to detect for most approaches.

As shown in the bottom axis, Alter-Re^2 on its own is able to detect the first anomaly, but misses both second and third ones. We must note that even though the signal is within the anomaly window at timestep 300 due to the approach of Lavin and Ahmad [20], the spike for which the anomaly is labelled only arrives around one hundred timesteps later. Recall thus becomes $\frac{1}{3} = 0.33$, while Precision is 100%, as the only signal raised was correct. The reasons for missing the third anomaly are similar to those explained above for Figure 5.1a.

Enabling pre-processing, however, has a great benefit. Figure 5.2b shows in the bottom axis that this way the third anomaly is also detected. Recall is therefore doubled to become $\frac{2}{3} = 0.667$. Although Precision is slightly decreased due to the one false positive signal at timestep 500, the aggregate metric, F-score is still increased by 60% denoting superior operation.

We would also like to point out that anomalies remain clearly visible for both (and in fact, most) datasets after pre-processing, fulfilling our aim entirely, and thus showing our hypothesis correct. As a result, even in the cases where no improvement is made in detection, rarely do we see any drops in performance.

Finally, we present aggregated results for all 58 NAB datasets for all six test cases (original and five pre-processing paarmeter settings). These are shown in Figure 5.3. The red, green and blue bars denote averages of the Precision, Recall and F-score metric respectively for the given test case. The black lines above and below the bars show the standard deviation of the same metric.

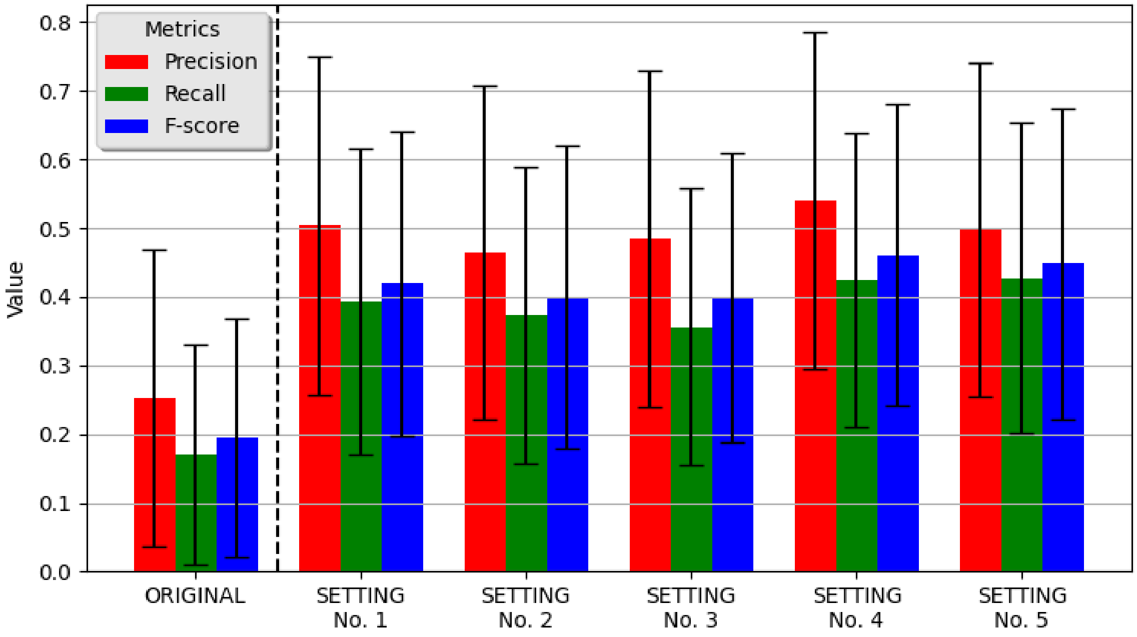


Figure 5.3: Alter-Re^2 run on original data compared to running on data from our pre-processing procedure on all NAB datasets

On the left side of the vertical dashed line, we show the results for Alter-Re^2 run on original data. On its right, we draw values for Alter-Re^2 run on data from our pre-processing procedure for all five parameter settings of VMD. It is clear from first glance that our new approach outperformed Alter-Re^2 from the viewpoint of every metric.

Even the worst performing Setting, No. 3, achieved 92% increase in Precision, 110% increase in Recall and 105% increase in F-score, effectively doubling the algorithm's performance. Setting No. 4 achieved the highest metric values with 0.54 Precision, 0.42 Recall and 0.46 F-score. These result in 113%, 150% and 137% improvement respectively when compared to running on original data, which had values of 0.25 Precision, 0.17 Recall and 0.19 F-score. Finally, compared to the original baseline, the average of all Settings achieved 97%, 133% and 118% performance upgrades in the respective metrics.

Another fortunate consequence of these results shown in Figure 5.3 is that performance is significantly less sensitive to VMD hyperparameters like the number of modes (K) and the bandwidth constraint coefficient (α) as we had previously expected.

Between the best performing pre-processing Setting, No. 4, and the worst, Setting No. 3, there is very little difference in our evaluation metrics. No. 4 only exceeds No. 3 by 11% Precision, 19% Recall and 16% F-score. We can thus conclude that their choice within reasonable bounds discussed earlier remains a minor concern.

However, analysis of the results by hyperparameter value does lead to insight about optimal values. Comparing the values by the bandwidth constraint parameter α shows that the smaller, 100 setting yields 7%, 14% and 12% better values in the respective metrics than the $\alpha = 500$ setting. Furthermore, when it comes to the number of modes, K, the trend in our experiments was the more modes we separate the data into, the better the performance. $K = 8$ means a 7% improvement in the F-score metric compared to four modes, while selecting K to be 12 yields 10% better results on average in the same metric compared to $K = 4$ similarly. Evidently, Setting No. 4 achieved the highest results with parameter settings $\{\alpha = 100; K = 12\}$.

To conclude, we believe the examples outlined at the beginning of this section clearly demonstrate the improvements introduced by our pre-processing procedure. Furthermore, the aggregated results on all 58 NAB datasets show that our scaling and decomposition approach managed to double the performance of Alter-Re^2 on a wide selection of datasets in all three selected, well-established metrics.

5.3 Discussion and implications

Analysis of our results outlined in the previous section yields the conclusion that, with little regard to the type of data, employing our mode decomposition approach leads to improved performance. Although we did notice some performance drops in a small minority of cases, these were vastly outnumbered by those that benefited from our approach.

Whether this is the consequence of our datasets under examination is yet to be determined, and warrants future research we intend to do. Extending tests to other data sources from within the networking domain and further beyond is in the scope of our interest and plans. However, given the wide range of sources NAB acquired their data from, and the number of experiments we conducted, we must argue large-scale applicability of our results.

Another issue yet unanswered is whether similar performance improvements can be achieved with other algorithms. In future, we intend to evaluate the efficacy of our pre-processing procedure using other state-of-the-art anomaly detectors like the ones mentioned in [2].

We also aim to extend our research to multi-dimensional time series datasets. Taking into account the correlation between different types of information from the same data source (e.g., the temperature, CPU and port information of the same router), should lead to better performance than one-dimensional analysis on all datasets separately. This applies both to the anomaly detector itself and the pre-processing schemes applied. Anomaly detectors might employ a majority vote by data dimensions for signalling an anomaly, or might employ a weighted average of signals based on various criteria. As for pre-preocessing, effective dimension reduction while preserving signs of anomalies, for example, will lead to less resource-intensive operation, that is necessary for real-time operation.

Dealing with periodic and aperiodic time series from the same data source will continue to require mode decomposition and transformation algorithms. Our goal is to create a framework that can select an optimal approach for data transformation for each type of data, and can run in real-time. It might also need to reevaluate its decision periodically to maintain long-term performance.

Such a framework necessitates robust classification of datasets based on patterns of behaviour, statistical values and other details. An article written by a member of our research team, that has been accepted for publication [3], introduces classification of data types into four classes. The selection and calculation of these is motivated by the observed difference of Alter-Re^2 performance on different types of time series. We also plan to extend this classification to include more criteria, and also take into account pre-processing differences.

Finally, we will continue to evaluate other state-of-the-art pre-processing schemes and data transformation methods to contrast them with our scaling and mode decomposition approach. We will also evaluate more hyperparameter settings for VMD. Considering that best performance was achieved by the highest number of modes tested, we plan to increase K even further until we approach optimal values. At the same time, using smaller α values (thus allowing even larger bandwidths per mode) will also be tested. We will also make steps to allow our pre-processing procedure to run in real time in a windowed fashion, and evaluate approaches to achieve that.

To sum up, we believe that the effective doubling of Alter-Re^2 performance can be improved further by more experimentation. We will also extend our scope when it comes to pre-processing and transformation methods, datasets, data types, anomaly detectors, and even dimensions.

Chapter 6

Conclusions

In this study, we set out to improve the performance of our previously designed algorithm, Alter-Re^2, by extending its applicability into the periodic data domain. We aimed to achieve this via the theoretically well-grounded and established Variational Mode Decomposition algorithm preceded by MinMax scaling.

We hypothesised that by removing all mode functions from the scaled original dataset the remaining residue would have drastically reduced content of regular, periodic patterns, while still preserving most signs of anomalies. We argue our hypothesis was proven right based on our experiments.

Evaluating our pre-processing approach on 58 datasets from various sources, we found that Alter-Re^2 achieved a 118% increase in the F-score metric on average, effectively doubling its average performance. We argue this shows the wide applicability of our method and the power of mode extraction.

In future, we plan to analyse the approach further, experimenting with more parameter settings, datasets, pre-processing and anomaly detection algorithms. We are also going to incorporate multidimensional datasets from the same source to exploit correlation and achieve more efficient and precise anomaly detection. In the long term, we plan to build a framework of classifiers, pre-processors and anomaly detectors that can select appropriate transformation schemes and anomaly detection engines in real time based on the type and pattern of data under analysis.

Acknowledgements

I would like to thank my advisor, dr. Károly Farkas for his continued, regular support throughout the development of the procedure and the writing of this study.

I would also like to extend my gratitude to Dr. Tien Van Do for his valuable comments on algorithm implementation and the text of this study. I also owe him thanks for the test infrastructure without which I could not have achieved such an extensive number of results. I also greatly appreciate the office space he provided me with.

List of Figures

List of Tables

Bibliography

[1] Ming-Chang Lee, Jia-Chun Lin, and Ernst Gunnar Gran. ReRe: A Lightweight Real-Time Ready-to-Go Anomaly Detection Approach for Time Series. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 322–327, 2020. DOI: `10.1109/COMPSAC48688.2020.0-226`.

[2] Daniel Vajda, Adrian Pekar, and Karoly Farkas. Towards Machine Learning-based Anomaly Detection on Time-Series Data. *Infocommunications Journal*, XIII(1):36–44, 3 2021. DOI: `10.36244/ICJ.2021.1.5`.

[3] Karoly Farkas. AREP - an adaptive, machine learning-based algorithm for real-time anomaly detection on network telemetry data. *Neural Computing and Applications*, 2022. Accepted for publication.

[4] Konstantin Dragomiretskiy and Dominique Zosso. Variational mode decomposition. *IEEE Transactions on Signal Processing*, 62(3):531–544, Feb 2014. ISSN 1941-0476. DOI: `10.1109/TSP.2013.2288675`.

[5] Norden E. Huang, Zheng Shen, Steven R. Long, Manli C. Wu, Hsing H. Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H. Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1971):903–995, 1998. DOI: `10.1098/rspa.1998.0193`. URL `https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1998.0193`.

[6] Zhaohua Wu and Norden E. Huang. Ensemble empirical mode decomposition: A noise-assisted data analysis method. *Advances in Adaptive Data Analysis*, 01(01): 1–41, 2009. DOI: `10.1142/S1793536909000047`. URL `https://doi.org/10.1142/S1793536909000047`.

[7] María E. Torres, Marcelo A. Colominas, Gastón Schlotthauer, and Patrick Flandrin. A complete ensemble empirical mode decomposition with adaptive noise. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4144–4147, May 2011. DOI: `10.1109/ICASSP.2011.5947265`.

[8] Jérôme Gilles. Empirical wavelet transform. *IEEE Transactions on Signal Processing*, 61(16):3999–4010, Aug 2013. ISSN 1941-0476. DOI: `10.1109/TSP.2013.2265222`.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997. DOI: `10.1162/neco.1997.9.8.1735`.

[10] Tae Jun Lee, Justin Gottschlich, Nesime Tatbul, Eric Metcalf, and Stan Zdonik. Greenhouse: A Zero-Positive Machine Learning System for Time-Series Anomaly Detection, 2018.

[11] Ming-Chang Lee, Jia-Chun Lin, and Ernst Gunnar Gran. RePAD: Real-Time Proactive Anomaly Detection for Time Series. In Leonard Barolli, Flora Amato, Francesco Moscato, Tomoya Enokido, and Makoto Takizawa, editors, *Proceedings of the Advanced Information Networking and Applications*, pages 1291–1302, Cham, 2020. Springer International Publishing. ISBN 978-3-030-44041-1.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[13] S. Gopal Krishna Patro and Kishore Kumar Sahu. Normalization: A preprocessing stage, 2015. URL `https://arxiv.org/abs/1503.06462`.

[14] Mathias Johansson. The hilbert transform. *Mathematics Master's Thesis. Växjö University, Suecia. Disponible en internet: http://w3. msi. vxu. se/exarb/mj_ex. pdf, consultado el*, 19, 1999.

[15] Scikit-learn: preprocessing. MinMaxScaler, 2022. URL `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html`. Accessed: 2022-10-19.

[16] Vinícius R. Carvalho, Márcio F.D. Moraes, Antônio P. Braga, and Eduardo M.A.M. Mendes. Evaluating five different adaptive decomposition methods for eeg signal seizure detection and classification. *Biomedical Signal Processing and Control*, 62:102073, 2020. ISSN 1746-8094. DOI: `https://doi.org/10.1016/j.bspc.2020.102073`. URL `https://www.sciencedirect.com/science/article/pii/S1746809420302299`.

[17] PyPI. vmdpy, 2022. URL `https://pypi.org/project/vmdpy/`. Accessed: 2022-10-25.

[18] Numenta Inc. NAB: Numenta Anomaly Benchmark [Online code repository], 2022. URL `https://github.com/numenta/NAB`. Accessed: 2022-10-19.

[19] Numenta Inc. Numenta website, 2022. URL `https://numenta.com/`. Accessed: 2022-10-19.

[20] A. Lavin and S. Ahmad. Evaluating Real-Time Anomaly Detection Algorithms – The Numenta Anomaly Benchmark. In *Proceedings of the IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 38–44, 2015. DOI: `10.1109/ICMLA.2015.141`.