FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Assignment of master's thesis

Title:	Market Data Anomaly Detection
Student:	Bc. Eliáš El Frem
Supervisor:	Dr. Christian Mandler
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2025/2026

Instructions

Market risk is often defined as a risk of losses caused by market movements. Anomalies in market data (trade-related data such as price, bid/ask quotes, yield curves, and market volume for financial instruments and assets) often signalize the risk. A robust market data management process is a cornerstone of market risk management. Bianchetti & Scaringi (1), in their presentation at a QuantMinds International conference, showed the results of their research, in which they benchmarked various anomaly detection methods, such as Isolated Forests, autoencoders, and LSTM networks. The goal of this thesis is to conduct further research on the detection of anomalies in market data and to propose solutions suitable for use in bank systems for evaluating market risk.
Partial goals:

- Briefly discuss state-of-the-art methods for anomaly detection.
- Discuss the complexity of evaluating model performance in the absence of training/validation labels.
- Familiarize the reader with research in (1) and try to reproduce the results on chosen data to see if the results still apply to these.
  - If the results do not correspond, try to find a reason for this.
- Experiment with different hyperparameters to adjust the models to used data if needed.
- Compare the results from LSTM, autoencoders, and Isolated Forests on different asset classes.
- Observe if different models should model different asset classes.
- Find other models to benchmark the performance of the presented algorithms.

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 22 February 2024 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Bibliography

1. Learning Market Data Anomalies. Bianchetti, Marco and Scaringi, Marco. London : s.n., 2023. QuantMinds International.

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 22 February 2024 in Prague.

Master's thesis

ANOMALY DETECTION IN MARKET DATA

Bc. Eliáš El Frem

Faculty of Information Technology
Department of Applied Mathematics
Supervisor: Dr. Christian Mandler
May 9, 2024

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis: El Frem Eliáš. *Anomaly detection in market data.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2024.

Contents

List of Figures

List of Tables

List of code listings

I would like to express my most profound appreciation to my supervisor, Dr. Christian Mandler, for his invaluable guidance throughout the writing process of this thesis. I also express my sincere gratitude to all my family and friends, whose unwavering encouragement and support have been instrumental in completing this work.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Czech Technical University in Prague has the right to conclude a licence agreement on the utilization of this thesis as a school work pursuant of Section 60 (1) of the Act.

In accordance with Section 2373(2) of Act No. 89/2012 Coll., Civil Code, as amended, I hereby grant a non-exclusive authorisation (licence) to use this copyright work, including all computer programs and all their documentation (hereinafter collectively referred to as "the Work"), to all persons who wish to use the Work. Such persons shall be entitled to use the Work in any manner that does not diminish the value of the Work, and for any purpose (including use for profit). This authorisation is unlimited in time, territory and quantity.

In Prague on May 9, 2024

Abstract

This thesis studies anomaly detection in market data using advanced machine learning techniques. It evaluates the effectiveness of Long Short-Term Memory (LSTM) networks, Autoencoders, and Isolation Forests across various financial datasets, including equities, foreign exchange rates, and commodities.

The study conducts a comparative analysis of these models under different data transformations like linear detrending, differentiation, and relative change to determine their impact on anomaly detection. The research aims to extend the results of Bianchetti and Scaringi by introducing additional models, namely SVM and Hurst exponent, and comparing the performance across different asset classes.

Keywords Anomaly Detection, Market Data, Machine Learning, Unsupervised Learning, Neural Networks

Abstrakt

Tato diplomová práce se zabývá detekcí anomálií v obchodních datech za použití pokročilých technik strojového učení. Hodnotí efektivitu sítí dlouhé krátkodobé paměti (LSTM), Autoenkodérů a Izolation Forests napříč různými finančními datasety, včetně akcií, směnných kurzů a komodit.

Studie provádí srovnávací analýzu těchto modelů pod různými transformacemi dat, jako je lineární odstranění trendu, diferenciace a výpočet relativních změn, aby určila jejich vliv na detekci anomálií. Výzkum si klade za cíl rozšířit výsledky Bianchettiho a Scaringiho použitím dalších modelů, jmenovitě SVM a Hurstova exponentu a porovnáním výkonnosti napříč různými třídami aktiv.

Klíčová slova Detekce anomálií, Tržní data, Strojové učení, Učení bez učitele, Neuronové sítě

List of abbreviations

NN	Neural Network
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
kNN	k-Nearest Neighbors
IF	Isolated Forests
SVM	Support Vectaor Machine

Introduction

The financial markets are an intricate web of transactions influenced by many factors, from economic indicators to global events. In this environment, anomalies in market data are not only statistical outliers but signals that may indicate deeper systemic risks or rare, severe events. These anomalies can manifest as sudden spikes in trading volumes, unusual price fluctuations, or irregular shifts and can lead to significant financial disturbances. They often serve as precursors to larger economic disruptions, such as market crashes.

This thesis, inspired by the research of Bianchetti and Scaringi [1] where the authors highlighted the effectiveness of machine learning models, namely Isolation Forests, Autoencoders, and LSTM networks, in identifying market anomalies, aims to delve deeper into the realm of anomaly detection within financial markets. By exploring various machine learning techniques, this work seeks to extend the current understanding and application of anomaly detection methods.

Goals of this thesis

This thesis aims to implement the algorithms mentioned by the foundational research of Bianchetti and Scaringi and benchmark them along with other unsupervised learning algorithms. The author will then examine how transformations of the datasets influence the performance of the anomaly detection algorithms. In the end, according to his findings, the author will compare the selected algorithms and propose solutions for their usage on real-world data.

Chapter 1

Anomaly detection methodology

Anomaly detection is a problem tackled across various fields of industry and science, including medicine[2], cybersecurity[3], big data processing[4], Internet of Things[5]. This chapter will delve into the various approaches to address this problem. First, in 1.1, we will go through various tools to solve this problem. Subsequently, we will concentrate on the specific methods that will be utilized in this thesis, which are outlined in 1.3.

1.1 State of Art

To understand the various methods used for anomaly detection, we must first define what an anomaly is. One of the most referenced definitions is the one from D.M. Hawkins, which states that an anomaly is *"an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism"*[6]. Another well-known definition states that *"An outlier is an observation that appears to deviate markedly from other members of the sample in which it occurs."*[7] These and many more give us an idea of what we are searching for - samples that diverge from the others. The first definition also gives us a reason to search for these anomalies. They can often help us uncover underlying processes or outside influences that interfere with the process we are observing. There are many ways to tackle this problem. The basic solution to this problem is using deep learning, such as LSTM or Autoencoder [8] or just simple RNNs [9]. Wang et al. suggest an improvement in the form of the usage of Frequency-enhanced Conditional Variational Autoencoder.[10] Other tool from the neural network realm is the

Convolutional NN that can also be used for anomaly detection.[11] Munir et al. also proposed an alternative approach to the anomaly detection by fusing the statistical and machine learning approaches in their FusaAD technique.[12] Another approach could be by using the Extreme Studentized Deviate Test with seasonality components. [13] Other approach is to use One-Class SVM introduced in [14]. Another popular tool is the Isolation Forest. [15][16]

This thesis will make use of several of these tools, which will be in detail explained further.

1.2 Previous research

This thesis is based on the research of Bianchetti et al. from [1]. To summarize their research, In their research,the authors conducted a study focused on detecting anomalies in market data using machine learning models. The motivation behind this research is to understand how anomalies are able to impact market risk measures which rely on the distribution of profits and losses across various market scenarios. The authors note that real trading portfolios include a vast number of financial instruments across different asset classes.

As for hte experiments, the authors chose several unsupervised machine learning models: Isolation Forest, Autoencoder, and LSTM networks. These models are applied to various types of market data across different asset classes and dimensions. The dataset used in their study consists of historical series of interest rate yield curves with different trenors.

In terms of results and performance, the research finds that LSTM is very appealing thanks to its ability to consider both the entire market data structure and historical information. The study also compares different retraining strategies for their models, including static, sliding window, periodic, and triggered retraining, to determine the most effective approach in real-time applications.

The authors also tried to inject artificial anomalies into the data. This approach helps them evaluate the performance without having the labeled data prior. They observe that the Autoencoder model shows superior performance in identifying these injected anomalies compared to other models.

In this thesis, author re-implemented the used solution for freely accessible data described in Chapter 3, along with the experiments for artificial anomalies (although instead of injecting them, the author of this thesis creates whole new artificial time series and injected the anomalies into it).

1.3 Methods used in this thesis

This section will cover the various methods employed in this thesis. In Section 1.3.1, we will talk about two types of neural networks used. Then, in 1.3.2, we will study Isolation Forests, a widely-used technique for identifying outliers within a dataset. Section 1.3.3 will be dedicated to KNN, commonly used to separate the data. 1.3.4 features an unordinary way of tackling the anomaly detection problem, the usage of Hurst Exponential.

1.3.1 Neural Networks

Neural networks are machine learning models inspired by an organic brain. The network consists of nodes layered in one or multiple layers. Each node, usually called a neuron, connects to others through links representing synapses in a biological brain. These connections are weighted, and during the learning process, the weights are adjusted to improve the network's ability to make predictions or classifications based on input data. This adjustment is facilitated through a process known as backpropagation, where the network learns from errors by propagating them backward through the layers and adjusting the weights accordingly. Sample visualization of the neural network with three layers is in Figure 1.3.1.

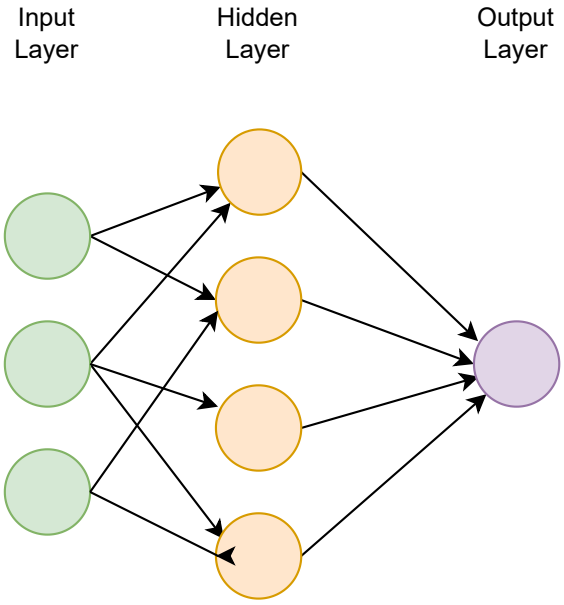

Figure 1.3.1 Sample neural network with three layers.

RNN is a particular case of a neural network used to imitate human brain

behavior better. The problem with traditional neural networks is that the input is processed only one way during classification or regression. However, the human brain always works with some previous knowledge. In neural networks, this is handled in RNNs by using their internal state (or hidden layers) to process sequences of inputs, allowing them to utilize information from prior inputs in their computations. This approach simulates the way the human brain recalls previous knowledge to inform future decisions, useful in tasks such as time series prediction[17], natural language processing[18], and speech recognition[19]. However, this ability to preserve information across time steps also introduces challenges like the difficulty in learning long-range dependencies within input sequences due to issues such as vanishing or exploding gradients.

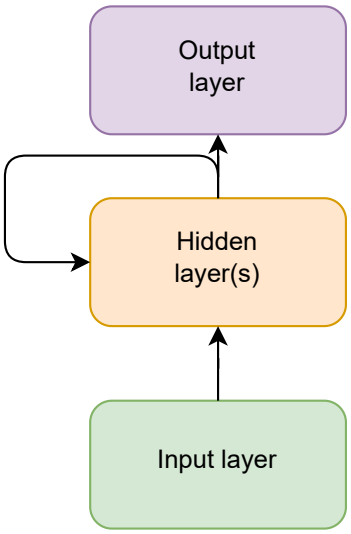

Figure 1.3.2 Visualization of a recurrent neural network.

A vanishing gradient problem occurs when the gradients of a network's weights become exponentially smaller as they propagate back through the layers during training. This problem effectively prevents the model from searching for a better model setup. This problem is solved by LSTM, which will be discussed in Section 1.3.1.1.

In the following two sections, we will introduce the LSTM and Autoencoder. These two NNs were selected by previous research as most suitable for solved tasks.

1.3.1.1 LSTM

The LSTM stands for Long short-term memory, a specialized type of RNN, specifically designed to process time series, that was originally introduced in

[20]. As stated in the previous section, the LSTM helps us overcome the vanishing gradient problem and is superior to traditional RNNs in terms of processing long time series with backward connections. At the writing of this thesis, LSTM is considered a state-of-the-art method for time series processing, although there are proposals to use Multi Recurrent Neural Networks instead [21].

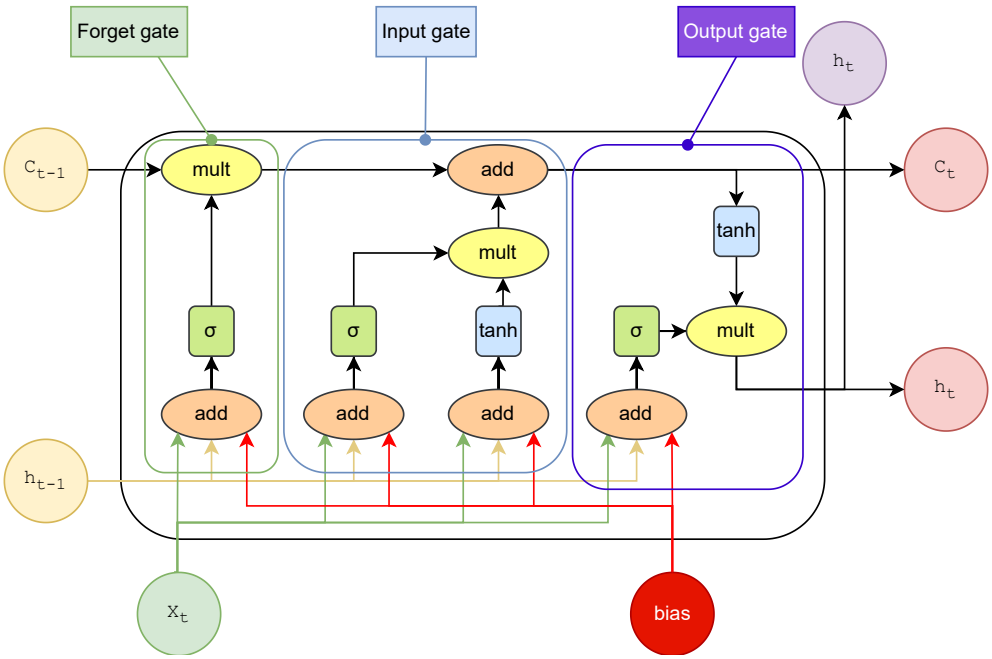

Figure 1.3.3 Visualization of an LSTM memory cell.

The overall design of LSTM helps it not only avoid vanishing gradients but also selectively remember patterns over long time intervals, making it ideal for applications like time series analysis. The main foundation of the design is a so-called memory cell. The LSTM memory cell consists of three parts - input gate, forget gate, and output gate. For the purposes of this thesis, we will use the definitions from the original authors in [20]. We will denote the functions representing these gates as i_t,f_t, and o_t, respectively.

The inputs for the LSTM memory cell are the following:

- x_t - feature input vector to the LSTM
- h_{t-1} - hidden state vector, the output of the previous cell,
- c_{t-1} - cell state vector, also the output of the previous cell.
- b - bias

Before h_{t-1} and x_t enter the LSTM memory cell, they are multiplied by weight vectors W_t and U_t respectively. The functions for the respective gates are then:

$$f_t = (\sigma(x_t W_f + h_{t-1} U_f + b)) \tag{1.1}$$

$$i_t = (\sigma(x_t W_i + h_{t-1} U_i + b) \times \tanh(x_t W_i + h_{t-1} U_i + b)) \tag{1.2}$$

$$o_t = (\sigma(x_t W_o + h_{t-1} U_o + b) \times \tanh(x_t W_o + h_{t-1} U_o + b)) \tag{1.3}$$

The outputs are then defined as:

$$c_t = (c_{t-1} f_t + i_t) \tag{1.4}$$

$$h_t = (\tanh c_t \times o_t) \tag{1.5}$$

h_t and c_t are then sent to the next cell as an input, with h_t also being the output of the current cell. This design helps us not only to avoid vanishing gradients but also to selectively remember patterns over long time intervals. This makes it great for anomaly detection on large time series. We can teach the LSTM on the data and then try to reconstruct it. If done correctly, LSTM should be able to more or less reconstruct the input series, with the exception of outliers, which we will then be able to detect.

1.3.1.2 Autoencoders

An autoencoder is defined as *a type of algorithm with the primary purpose of learning an "informative" representation of the data that can be used for different applications by learning to reconstruct a set of input observations well enough.*[22] They were first introduced in [23] as a neural network for reconstructing the input. [24]

Sample architecture of an Autoencoder can be found in Figure 1.3.4. By compressing the data in lower-dimensional space during encoding and then reconstructing it in the encoding part, they can learn efficient representations of data. We can then use a reconstruction error as an indicator of anomalous behavior. Since normal data have low reconstruction errors, anomalies will be significantly higher. The reconstruction error is computed as:

$$RE = ||x - \hat{x}||, \tag{1.6}$$

x being the input data and $\hat{x}$ being the reconstruction.

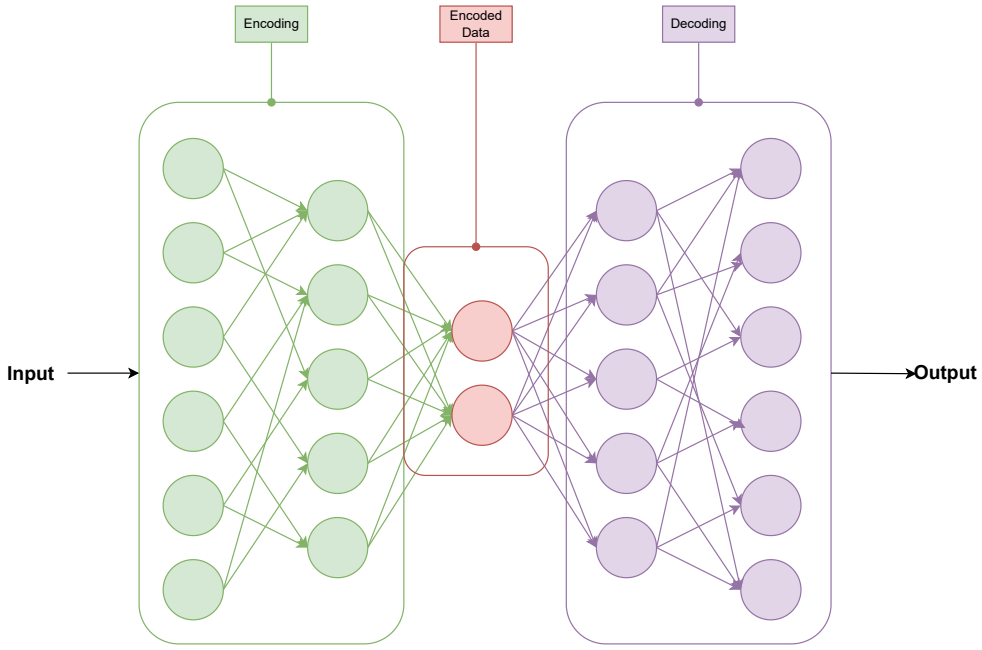

Figure 1.3.4 Visualization of an Autoencoder network.

1.3.2 Isolation Forests

Isolation Forests, first introduced in 2008 in [15], is an unsupervised learning model made specifically for anomaly detection. It operates with the premise that the anomaly points differ from each other and are not numerous. Therefore, they can be easily isolated from others.

The algorithm works by randomly splitting the data - in each step, we select a feature randomly and then choose a random split value within the minimum and maximum values of that feature. This leads to the creation of binary trees, known as Isolation Trees (iTrees).

Each iTree in the forest is built by recursively partitioning the data. At every step, the algorithm picks a feature and a split point randomly, continuing this process until the tree reaches a maximum height, which is set as hyperparameter, or until all the data points at a node are identical or only one point remains. The path length from the root of the tree to a node indicates the number of splits required to isolate a sample.

An Isolation Forest consists of multiple iTrees, with the number of trees being also a hyperparameter. For each iTree, the algorithm uses a random sample of the dataset, which allows the algorithm to be both efficient and effective in detecting anomalies.

The isolation of a point is measured by the path length from the root to the

leaf. This path length is then averaged over all iTrees to determine an anomaly score for each data point. Anomalies are characterized by shorter path lengths, as they are isolated quicker than normal points. After this, points with shorter average path lengths are identified as anomalies, typically indicated by scores close to 1, while normal points have longer path lengths, resulting in scores closer to 0.5 or lower. It is commonly used in fraud detection. For the purpose of this thesis we will use scikit-learn library implementation. [25]

1.3.3 SVM

One-Class SVM is an unsupervised learning algorithm that is trained only on the normal data. It learns the boundaries of these points and is therefore able to classify any points that lie outside the learned region as anomalies. This approach is particularly useful in situations where you have a lot of normal data but very few anomalous examples. The idea behind is to fit the smallest possible sphere (in feature space) around the data. Then the model will learn what is "normal" and what is an outlier. One-Class SVM was first introduced by Schölkopf at al. in [14], from where the following definitions have been taken. The objective of One-Class SVM is to fit the data and separate it from the origin in the feature space, maximizing the distance from this hyperplane to the origin. The problem can be formulated as:

$$\min_{\mathbf{w},\xi,\rho} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho$$

subject to

$$(\mathbf{w}\cdot\phi(\mathbf{x}_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad i = 1,\ldots,n$$

where $\phi(\mathbf{x}_i)$ maps the input data $\mathbf{x}_i$ and where where: $\mathbf{w}$ is the weight vector in the feature space, ρ is the offset of the hyperplane from the origin, $\phi(\mathbf{x}_i)$ maps $\mathbf{x}_i$ into a higher (possibly infinite) dimensional space, ξ_i are slack variables allowing some points to be on the wrong side of the hyperplane, $\nu \in (0, 1]$ is a parameter that controls the trade-off between maximizing the distance of the hyperplane from the origin and minimizing the number of margin errors (points for which $(\mathbf{w}\cdot\phi(\mathbf{x}_i)) < \rho$).

A common kernel is the Radial Basis Function (RBF):

$$K(\mathbf{x}_i,\mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

where γ is a parameter that defines the influence of a single training example.

After training, the decision function is:

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i) - \rho\right)$$

As in the case Isolated Forest, we will use the scikit-learn library implementation of this algorithm. [25]

1.3.4 Hurst Exponential

The Hurst exponent, named after British hydrologist Harold Edwin Hurst, is a statistical tool for determining the long-term memory of time series data. Therefore, it can help us quantify the long-range dependence and persistence of the underlying processes behind the time series.[26] It has usage in multiple fields besides finance [27], such as medicine [28] and meteorology [29]. The Hurst Exponential can be calculated in multiple ways, such as using detrended fluctuation analysis [30] or ranged scale analysis introduced in cite-Hurst1951LongTermSC. In this thesis, we will use the second method and will define it below. For our time series x, we can define the mean-adjusted series as :

$$X_t = x_t - \bar{x}, for t = 1, 2..., n,$$

where $\bar{x}$ is the sample mean of the time series and where n is the length of the time series.

Then we calculate the cumulative sum of the mean-adjusted data:

$$Y_t = \sum_{i=1}^{t} X_i,$$

for $t = 1, 2, \ldots, n$.

After that, we can compute the range $R(n)$ of the cumulative deviations as

$$R(n) = \max_{1 \le t \le n} Y_t - \min_{1 \le t \le n} Y_t.$$

After that, we can the standard deviation $S(n)$ of the original time series:

$$S(n) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

The rescaled range is then given as the ratio of the range $R(n)$ to the

standard deviation $S(n)$:

$$\frac{R(n)}{S(n)} = \frac{\max_{1\leq t\leq n}\sum_{i=1}^{t}(x_i - \bar{x}) - \min_{1\leq t\leq n}\sum_{i=1}^{t}(x_i - \bar{x})}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

Than the Hurst exponent in:

$$E[\frac{R(n)}{S(n)}] = Cn^H, n \to \infty,$$

where C is a constant and H is the required Hurst exponent.

We will try to use the Hurst Exponent as suggested in [27] to help us detect anomalous market behaviour. The implementation will be taken from https://github.com/Mottl/hurst/blob/master/hurst/__init__.py. [31]

Chapter 2

Unsupervised learning evaluation

Unsupervised learning is a machine learning paradigm where algorithms learn patterns from unlabeled data without explicit instruction. This gives us the possibility to teach our models on unlabeled data, or data for which we do not know what the features determine them. As powerful tools as they are, the fact that we do not have labels also means that we have no reference point to determine if the model is performing well in terms of generating the correct output since traditional measures do not apply here due to the lack of labels. Therefore, we have to employ alternative methods to estimate the performance and compare the models. Aside from the problems mentioned with validation, we also often have a hard time interpreting the results, as they are not as clear as those for supervised learning. Another challenge we are facing when using unsupervised methods is the susceptibility to noise in data - the models search for patterns to adapt, and if they find some patterns in noise, they can adapt to it instead of the patterns in the features. In the following sections, we will go through the current state of art methods, as well as methods used in this thesis.

2.1 State of Art methods for unsupervised learning evaluation

One of the most common methods for unsupervised learning is clustering - separating data into clusters based on their learned features. If we want to evaluate the performance of a clustering algorithm, we may look at how well it separates into classes. For this purpose, we may use Silouhette analysis, a

metric used to evaluate the quality of data separation. It measures the separation between classes and the cohesion within class clusters.[32] This algorithm can also be used on big data databases, making it feasible for evaluating the outputs of a large number of models.[33] Another method for separation evaluation is the Daves-Bouldin index, which measures the average 'similarity' between clusters - the ratio of within-cluster scatter to between-cluster separation.[34] This algorithm is usually used to evaluate the quality of clustering algorithms when no labels are available. The third algorithm for cluster separation quality assessment is the Calinski-Harabasz Index. This algorithm scores clusters by the ratio of the sum of between-clusters dispersion to within-cluster dispersion. The advantage of this algorithm is that it can also show us the density of clusters. [35] For other methods like principal analysis componetns[36] or autoencoders[37] that use dimensionality reduction to increase the interpretability of the data, the quality of the dimensionality reduction can often be evaluated based on the reconstruction error. This method is also used for anomaly detection. [38] In the case of having mixed unlabeled samples with labeled ones, we can use pseudo-labeling by designating certain data points as anomalies based on the model's internal criteria (e.g., deviation from a norm or threshold).[39] These pseudo-labels can then be used to calculate metrics like precision and recall. It also allows us to use the ROC curve to measure the ability of the model to distinguish between anomalous and non-anomalous samples. [40]

2.2 Methods used in this thesis

In the following section, we will briefly go through all methods for evaluation of the unsupervised algorithms that were used for experiments.

2.2.1 Visual

Visually checking the results may seem like a tedious task, but in reality, we just want to confirm that the model reacts to events we know were anomalous via expert knowledge. Such events are historical crises, war conflicts, natural disasters, and regime changes. Functional models should be able to identify these events since they are usually responsible for visible market shifts. In the experimental part, we will describe the graphs and talk briefly about the points where the anomaly should have been detected, but was not. This is not a quantitative measure, but rather a subjective evaluation metric that is used to determine to which data is the algorithm best match.

2.2.2 Separation

We can also look at the quality of separating anomalous and non-anomalous data. For this, we will use a so-called Silouhette score, which measures how similar an object is to its own cluster compared to other clusters. A high silhouette score indicates that the clusters are well separated and that the data within a cluster is similar.[32] Silouhette score for i^{th} point is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}, \tag{2.1}$$

where $a(i)$ is the mean intra-cluster distance for the i^{th} point and $b(i)$ is the mean nearest-cluster distance for the i^{th} point.

2.2.3 Reconstruction eror

Reconstruction error[38] in unsupervised learning helps us to quantify the difference between original input data and its reconstruction after processing through our models. This metric measures how well the model can replicate the input data. This way, we can get a direct measure of how much information is lost in the model's process, which is crucial for tasks like feature extraction and data compression. This helps us to compare their effectiveness in preserving data integrity and detail. We will mention the reconstruction error as the part of visual check, rather than quantifying it as a number.

2.2.4 Coresponding with other models

Another measurement we can do is to look at how the anomaly methods identify and correspond with each other. This can give us insight into how effective the methods were in identifying anomalies that other models confirm. It can also tell us how different models could work together. This is equivalent to confusion metrics, but since we do not know the labels beforehand, we will use other algorithms to provide us with labels.

2.2.5 Artificial anomalies

This method helps us to convert the unsupervised task to a supervised one by giving us the possibility to evaluate algorithms on known data. The approach that will be used in this thesis is generating artificial random noise with pre-determined anomalies. We will create a random walk with trend and seasonality and then insert anomalies and random shifts into it so it resembles the financial data.

Chapter 3

Data used

This chapter deals with data used during the experiments. For the financial data, we wanted to cover different financial products due to their different properties. All of the data used were obtained from https://finance.yahoo.com/.

3.1 Equities

Equity rates (stock prices) are prices at which we can buy and sell stocks of individual companies. They are sensitive to a myriad of factors - corporate earnings, market sentiment, etc. These rates are highly sensitive to factors like corporate earnings and market sentiment, exhibiting significant volatility compared to interest rates. This volatility impacts investment portfolios and company valuations. The associated risks can be diminished using financial instruments, for example:

- Equity (stock) option - contract granting the buyer the right (but not obligation) to buy/sell a stock for a pre-determined price,
- ETFs (Exchange Traded Funds) - investment funds holding multiple assets (usually equities, but can also contain commodities) that can be traded like stocks and aim to mirror the performance of a specific index or industry sector. They are usually balanced, so they hold their value.

These instruments have their downsides; for example, options come with extra fees that are usually paid to the counterparty. These fees can be adjusted by observing the market, especially when we detect some anomalous behavior. Another reason to analyze them is to adjust one's trading strategies, as well as balance the ETFs mentioned above.

For the purpose of anomaly detection research, I have chosen two equity products. First is the Standard and Poor's 500 index, commonly denoted as just S&P500 or SP500, which is a well-known stock market index that measures the performance of 500 large US stock exchange-listed companies. I have chosen this product since it reflects the overall health and performance of the US economy, featuring multiple industry sectors. The second product is Apple Inc. (AAPL) stocks, which are interesting due to the fact, that Apple is known for its robust financial health and innovation. Apple's market capitalization has a notable impact on S&P500, so it can be interesting to compare these two products in terms of anomalies. From a mathematical point of view, the difference between the relative growth of AAPL and S&P500 can be a good benchmark for various models.

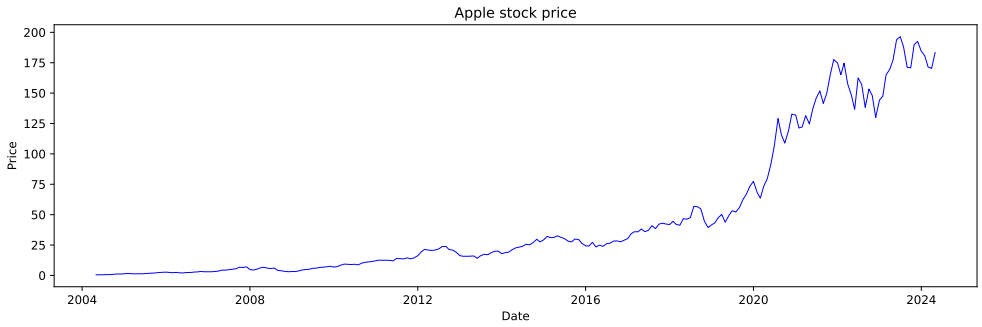

Figure 3.1.1 Apple stock prices for last 20 years.

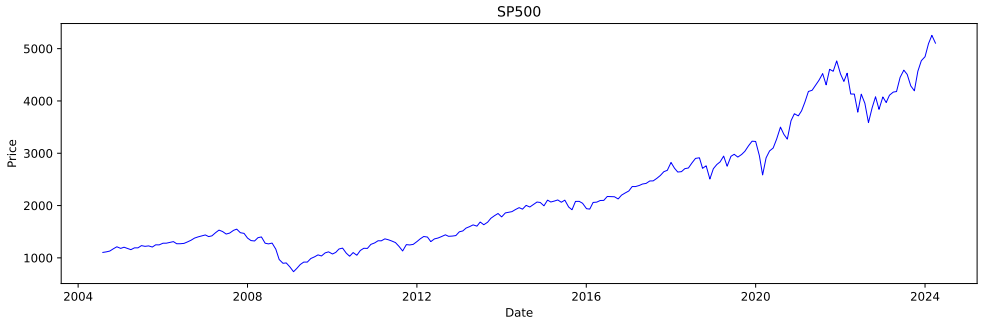

Figure 3.1.2 S&P500 prices for last 20 years.

3.2 FX Rates

FX Rates (foreign exchange rates) are simply rates for which we can exchange one currency for another. They are susceptible to many factors, for example,

political events, central bank policies, economic indicators, or changes in expectations for future interest rates. Unlike equities in Section 3.1, where we took into account the performance of individual companies, in this case, we are looking at the ”performance” of whole countries. While not as volatile as equities, fluctuation for these is still very high. They are carefully monitored by businesses worldwide due to their impact on the perceived prices of goods, especially in today's global markets [41]. Similarly to the equities, the risk in this market can be mitigated through various instruments, such as:

- FX forward - contract for money exchange for a pre-determined rate on a future date,
- FX option - similarly to equity options, we agree to a pre-determined rate, at which we are entitled, but not obligated to buy the currency,
- FX swap - contract involving two counterparties who agree to exchange specific amounts of one currency for another at an agreed rate on one date and then reverse the exchange at a different rate later. This enables effective currency exposure management and liquidity transitions between currencies.

These instruments again come with a price that can be adjusted by observing the market behavior. Identifying anomalies in FX rates can also provide insights into market behaviors that diverge from what the market hypothesis typically predicts. These deviations, particularly noticeable when reacting to news events, can highlight potential market inefficiencies. Such insights are crucial for developing effective trading strategies that capitalize on these anomalies since there exist trading strategies that generate abnormal profits by exploiting the detected anomalies.[42]

We have chosen these two Euro FX rate curves for the experiments - one for the Japanese yen and the other for the US dollar. This selection is based on several factors. First, we have significant differences in their absolute values. Typically, the EUR to USD rate hovers around 1, while the EUR to JPY rate exceeds 100. Furthermore, we have to take into account the impact of the Asian market on the EUR/JPY curve, which can lead to anomalous indicators due to sudden market shifts. Additionally, there are notable differences in the fiscal and monetary policies of the US, Eurozone, and Japan. The first two tend to stimulate the market more through lower interest rates, while Japan has a long-term deflationary environment.[43][44] The graphs for these two curves can be found in Figures 3.2.2 and 3.2.1.

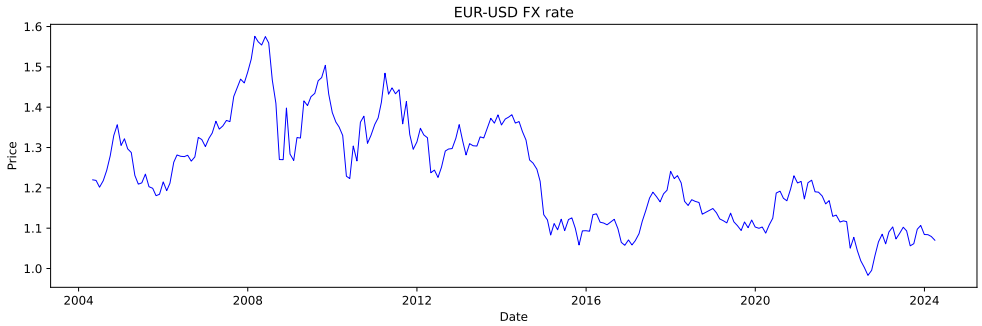

Figure 3.2.1 FX rate for EUR/USD for last 20 years.

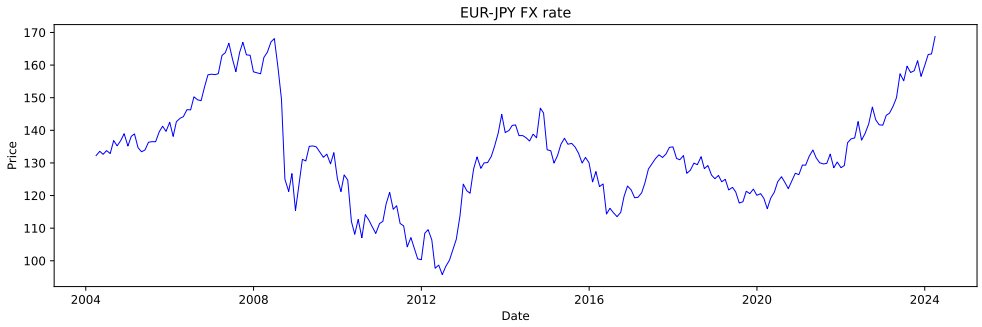

Figure 3.2.2 FX rate for EUR/JPY for last 20 years.

3.3 Commodities

Commodities represent real-world assets, such as gold, oil, platinum, etc. They are usually the most volatile financial products (aside from penny stocks and cryptocurrencies).

For the purposes of this thesis, I chose two commodities: oil (Figure 3.3.1) and gold (Figure 3.3.2). From the graphs, we can see that gold is much less volatile than oil and maintains steady growth. Oil, on the other hand, fluctuates often, reflecting anomalies in the market caused by conflicts in the Middle East and Eastern Europe.

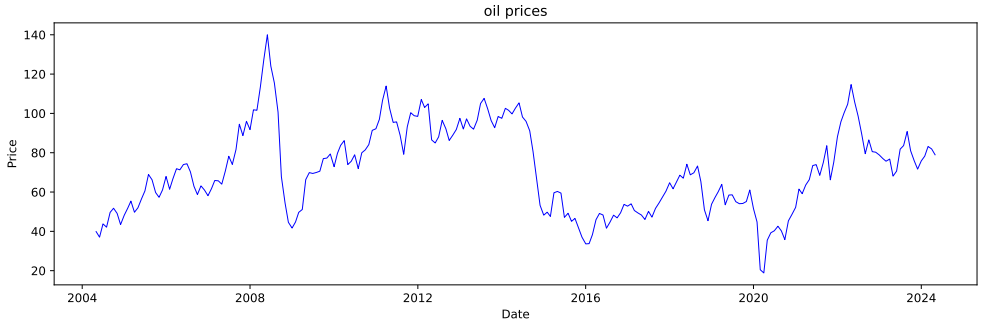

Figure 3.3.1 Oil prices for last 20 years.

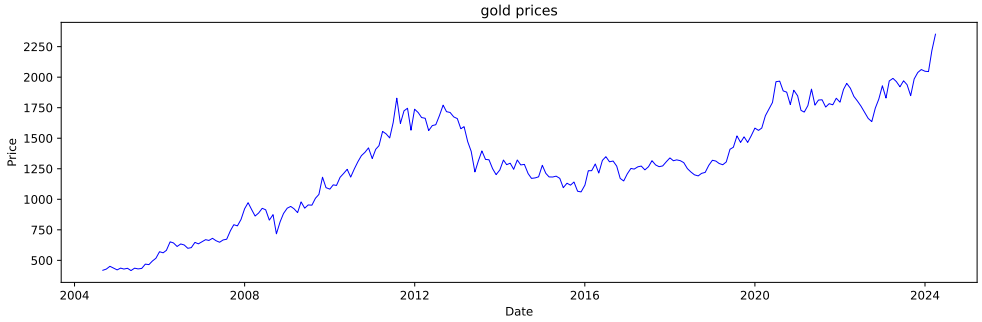

Figure 3.3.2 Gold prices for last 20 years.

Chapter 4

Experiments

This chapter is dedicated to the experiments in which the author will attempt to verify the results of the original research and then evaluate the methods used. Section 4.1 describes the transformation of the datasets we use to see how they improve/damage the detection. In Sections 4.2, 4.3, 4.4, 4.5 and 4.6, we will go through the graphs that show anomaly detection using algorithms described in Chapter 1 on datasets described in Chapter 3. For the time series, we will choose the last 20 years of monthly observation of the closing price (the price of the product when the market was closed). After that, in Section 4.7, the author will describe the attempt to reproduce the results of sequential retraining. Sections 4.8, 4.9 and 4.10 will be dedicated to evaluating the algorithms and their performance.

4.1 Data adjutments

We will describe the behavior of the algorithms on the data itself, and then we will modify the datasets in the following ways:

Linear detrend For some algorithms, the presence of the trend can be harmful. Therefore, we should try to get rid of it to improve the performance of these algorithms. We take time series y and calculate detrended series $\hat{y}$ as:

$$\hat{y} = y - (\beta_0 + \beta_1 t), \tag{4.1}$$

Where β_0 and β_1 are the intercept and slope, respectively, of the linear regression line fitted to the time series, and t is the index of the series.

Differentiation By differentiating the time series, we will see the difference between the observations. This could help better capture sudden shifts in the data that signalize the anomalies. The members of the differentiated time series Δy are defined as:

$$\Delta y_i = y_{i+1} - y_i, \tag{4.2}$$

where $y_i, i = 1, 2...n-1$ are members of the original time series y.

Relative change In the case of the data like Apple stocks in Figure 3.1.1, we can see that the values are rising significantly. Therefore, if there were some slight anomaly in the beginning, we would not register it as an anomaly, even though relatively at that time, it was anomalous. That can also happen the other way, where if we had large values at some point, the changes in small data would not be registered. For this reason, we will use the time series of relative changes p, defined as:

$$p_i = \frac{y_{i+1} - y_i}{y_i} \times 100\% \tag{4.3}$$

The anomalies will be projected onto the original time series for all data adjustments, while the reconstruction or isolation/outlier score will be projected on the adjusted time series.

4.2 Autoencoder anomaly detection

4.2.1 Normal data

Figures 4.2.1, 4.2.2, 4.2.3, 4.2.4, 4.2.5, 4.2.6 describe the behaviour of Autoencoder without editing the data. We can see that it mostly highlights the greatest and lowest values. The best example is Figure 4.2.1, which depicts Apple stocks, which have significantly risen in the last five years. We can see that the only anomalies highlighted are sudden shifts in the period from 2020 to 2024.

If we look at the reconstruction, the reconstruction in these years was rather bad, and thus, the false anomaly detections. FX Rates in Figures 4.2.2 and 4.2.3 also confirm this. However, the reconstruction is not so bad. This suggests that Autoencoder will work better with values that oscillate around one value, such as already mentioned FX rates and some sorts of commodities, such as oil, as shown in Figure 4.2.5, where the behavior is similar. S&P500 in 4.2.6 shows similar properties to Apple stocks, confirming that the trend is harmful for detection.

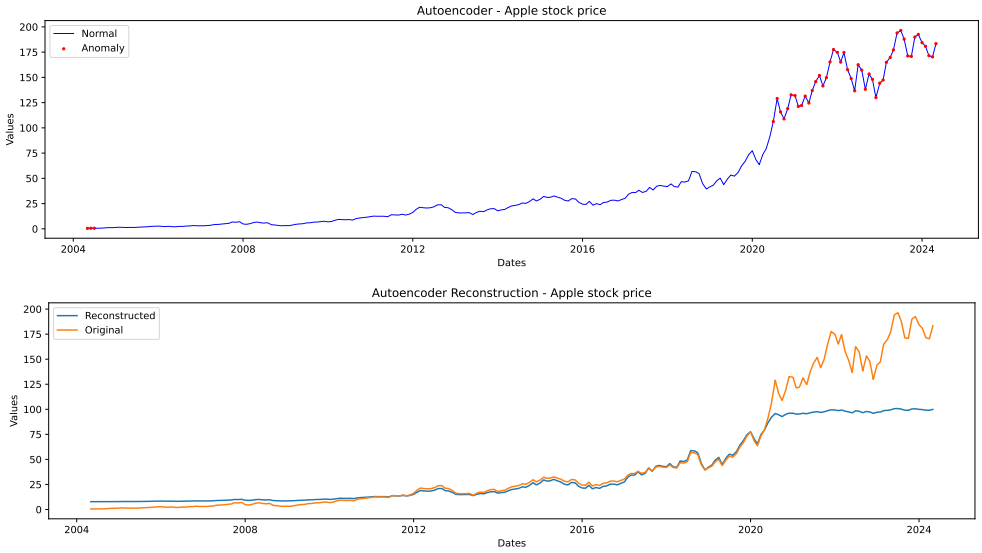

Figure 4.2.1 Apple stock price anomalies and reconstruction for Autoencoder.

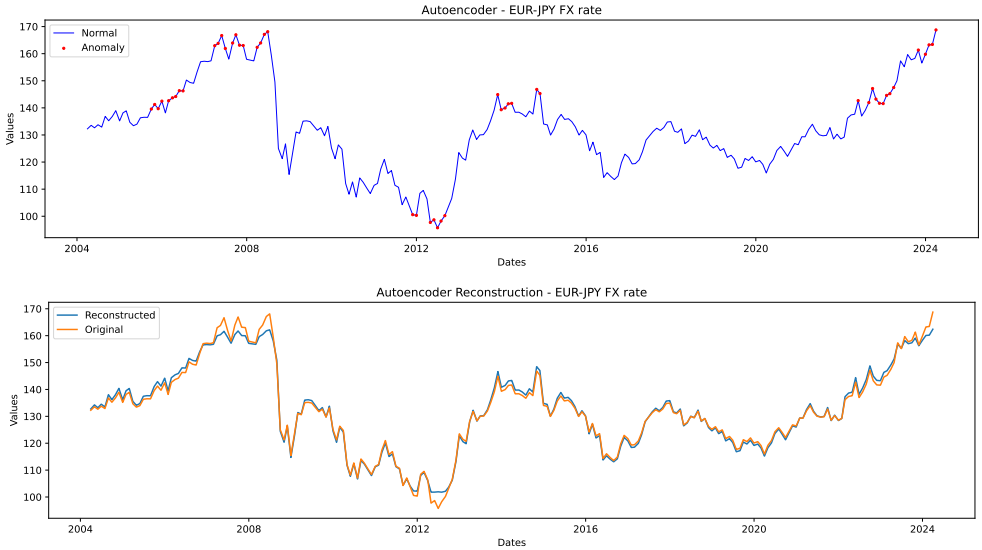

Figure 4.2.2 EUR/JPY FX rate anomalies and reconstruction for Autoencoder.

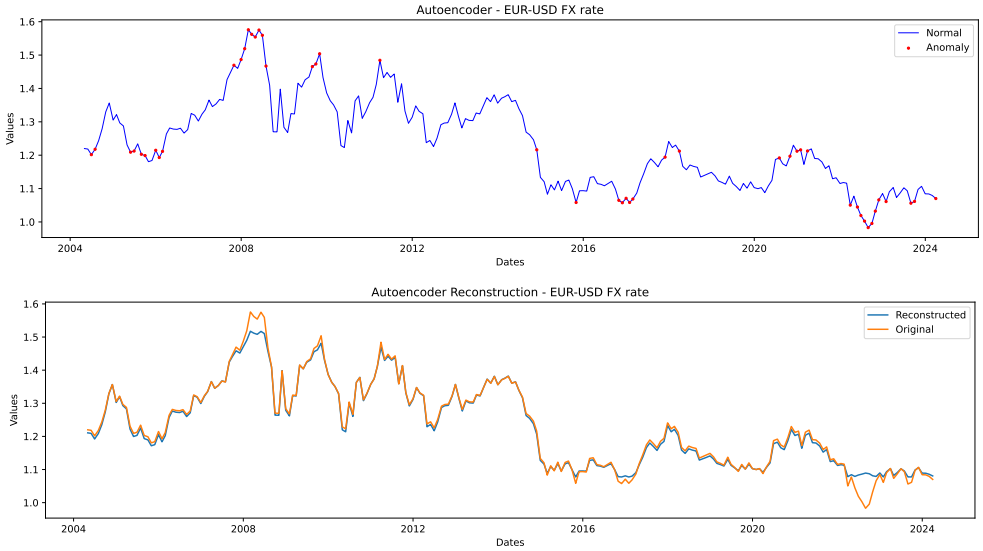

Figure 4.2.3 EUR/USD FX rate anomalies and reconstruction for Autoencoder.

An interesting case is gold in Figure 4.2.4, where the trend is not that steep, and the reconstruction is rather good, except for the end and beginning, which causes these places to be identified as anomalies.

Overall using Autoencoder on normal data did not bring us the best results, Apple stocks, S&P500 and gold prices ignored the 2008 market crash as well as some other crashes. We could see that the reconstruction was not perfect by far, and the only place where we got more or less satisfying results was the oil curve in 4.2.5. Here, the anomalies corresponded to major financial crises, like the one in 2008, 2015, 2018, and 2020, as well as the jump after the Russian invasion in 2022, where the prices went rapidly upwards. [45]

4.2.2 Detrended data

When using data where the trend was removed, the results improved in some cases. While looking at the Apple stock in Figure 4.2.7 and S&P500 in Figure 4.2.14, we can see that the data between 2020-2024 are fitted more correctly, although still missing a part. Compared to the previous section's Figure 4.2.1, we have correctly identified the anomalies in the Q4 of 2019. For the FX rates and oil prices, there was an insignificant improvement in 2008, where one more anomaly point is highlighted at the part of the graph depicting the market crash. Gold prices (Figure 4.2.10), on the other hand, show significant improvement, with the beginning and the end of the time series predicted correctly, the anomalies are more distributed around the graph, although we

Figure 4.2.4 Gold price anomalies and reconstruction for Autoencoder.

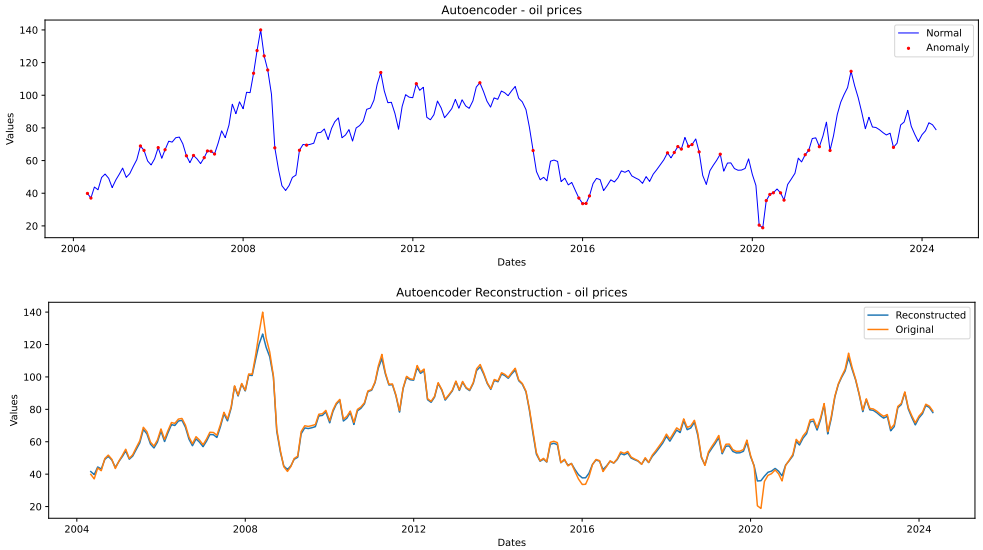

Figure 4.2.5 Oil price anomalies and reconstruction for Autoencoder.

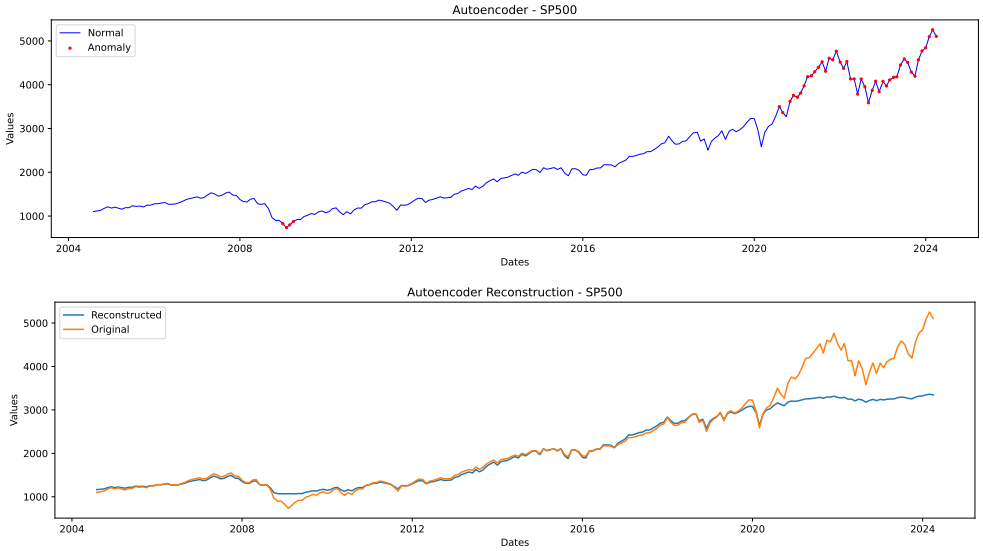

Figure 4.2.6 S&P500 index price anomalies and reconstruction for Autoencoder.

can see that the points are more concentrated around the most significant values and leave out some parts where anomalous behavior would be expected (for example 2008).

We can see that the improvement was negligible on this data, whereas the original time series worked well. For Apple stocks and S&P500 it worked better than before, but the results were still not satisfying. On gold, on the other hand, the Autoencoder worked very well and showed significant improvement.

4.2.3 Differenced data

The differenced data should provide us with better highlighting of the anomalous behavior by helping the Autoencoder to focus on the residuals or noise. Since the anomalies are defined by sudden changes rather than absolute values, the differenced data provide a good option for Autoencoder to identify the genuinely anomalous points. Since the Autoencoder does not rely on long-term dependencies, unlike the LSTM, the differentiation should not damage the learning process.

For the stocks, we can see that the anomaly detection for Apple stocks in Figure 4.2.13 is similar to Figure 4.2.1. The explanation for this behavior can be found in the reconstruction graph, which also shows the differenced series - we can notice that the absolute difference gets bigger in the 2020s, and therefore, the anomalies are only highlighted as sudden shifts in this time area. While the reconstruction is much better, the detection suffers from the same

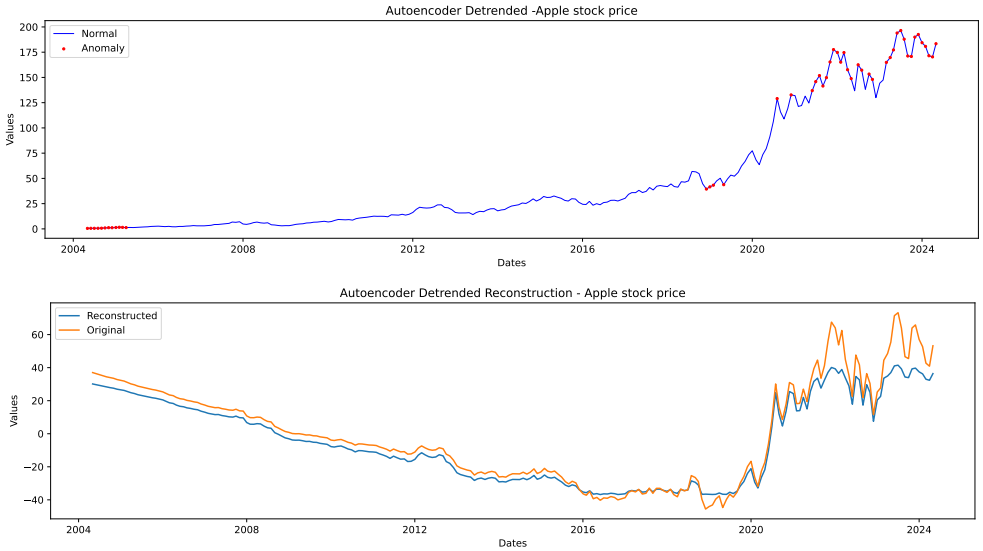

Figure 4.2.7 Apple stock price anomalies and reconstruction for Autoencoder with detrended data.

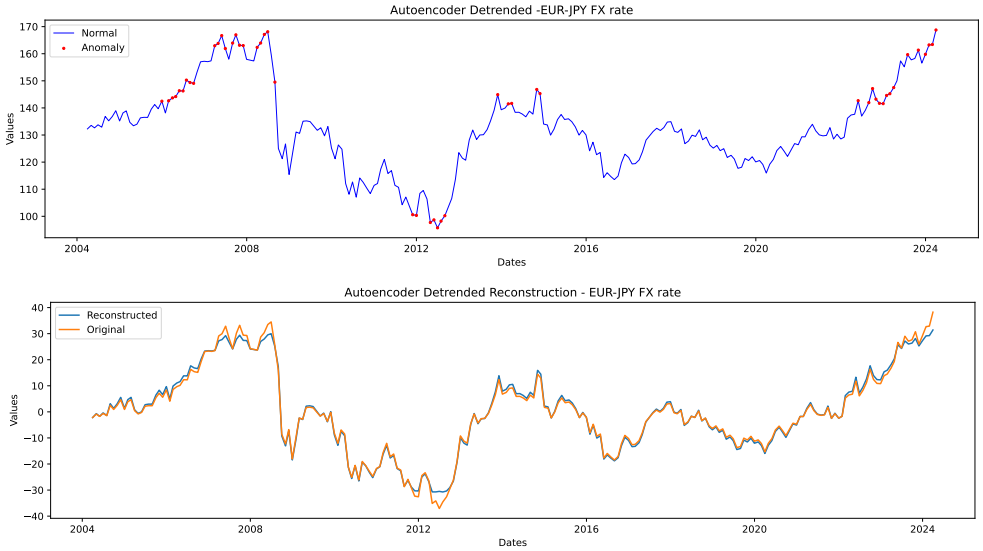

Figure 4.2.8 EUR/JPY FX rate anomalies and reconstruction for Autoencoder with detrended data.

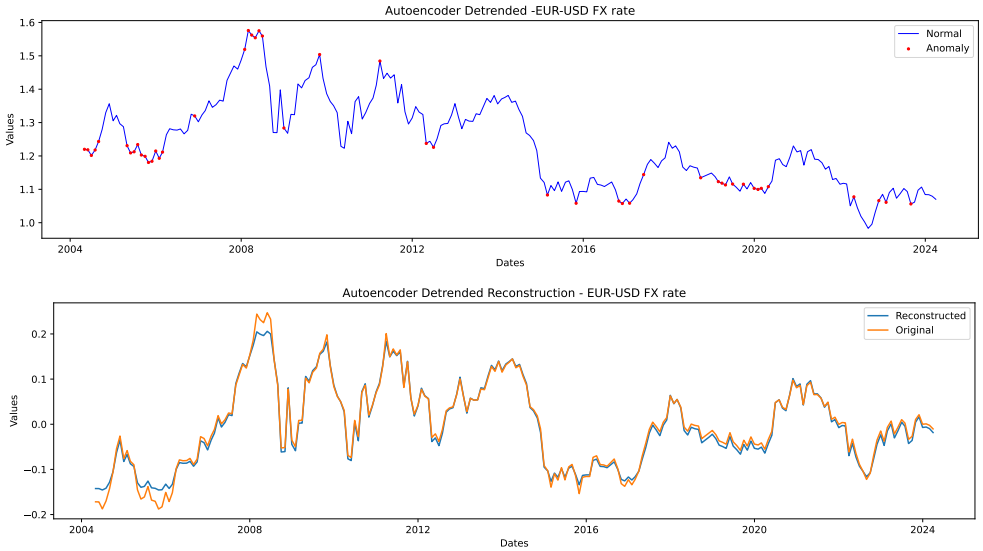

Figure 4.2.9 EUR/USD FX rate anomalies and reconstruction for Autoencoder with detrended data.

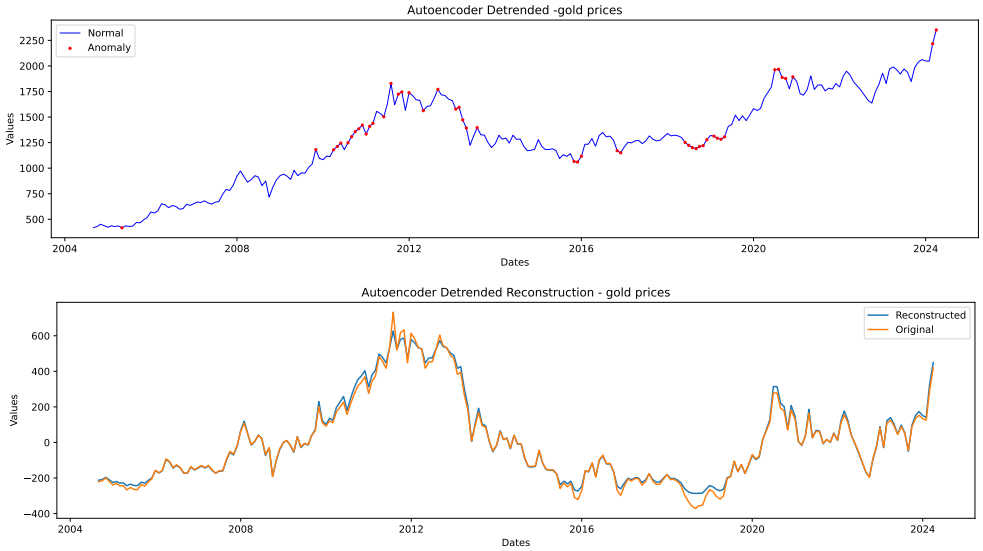

Figure 4.2.10 Gold price anomalies and reconstruction for Autoencoder with detrended data.

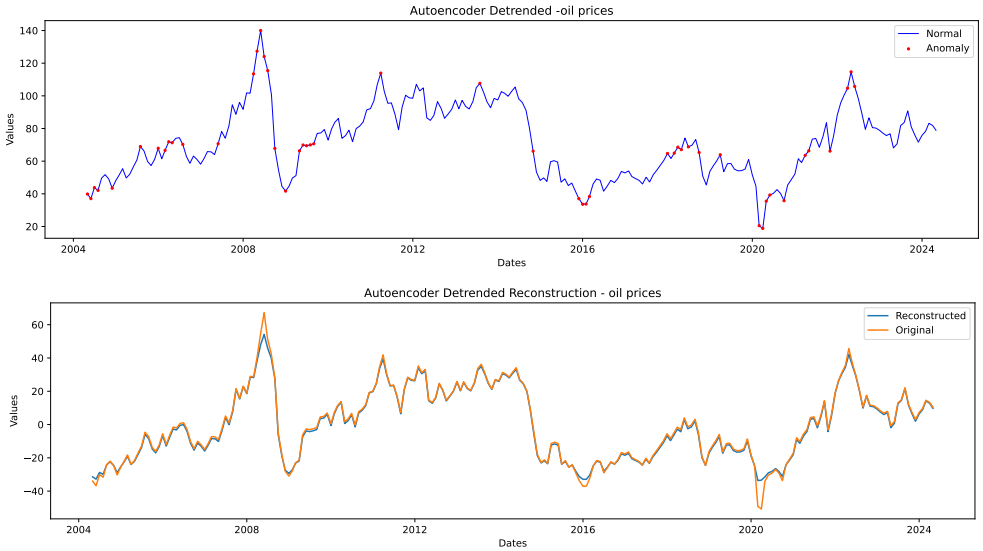

Figure 4.2.11 Oil price anomalies and reconstruction for Autoencoder with detrended data.

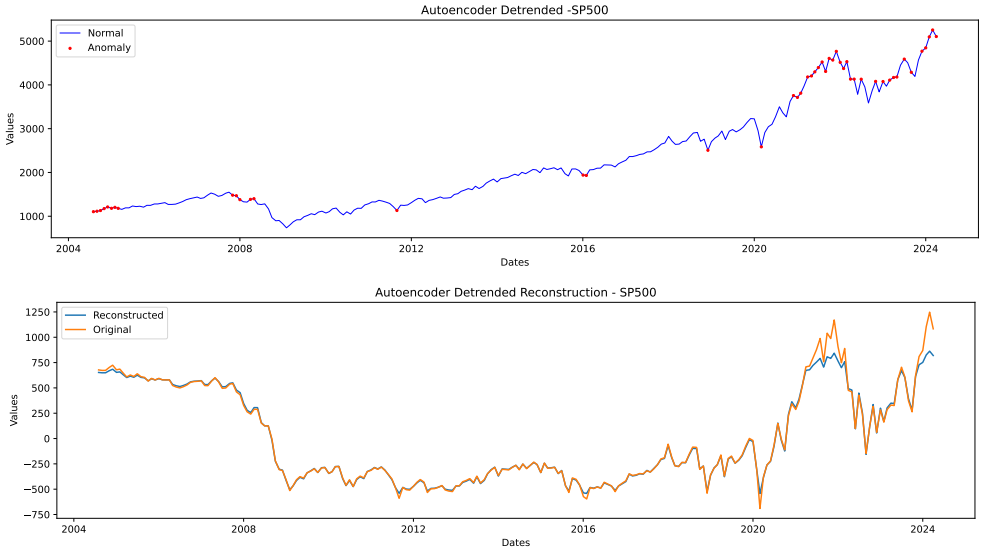

Figure 4.2.12 S&P500 index price anomalies and reconstruction for Autoencoder with detrended data.

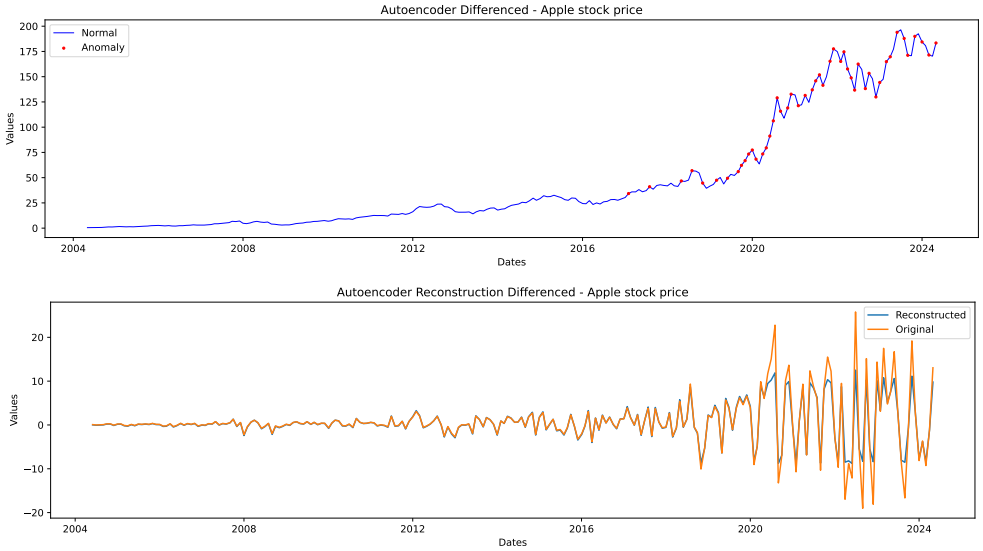

Figure 4.2.13 Apple stock price anomalies and reconstruction for Autoencoder with differenced data.

problems as with the original time series. The S&P500 has similar behavior; Figures 4.2.6 and 4.2.14 show similar findings, although we discovered some other anomalies that were not after the year 2020, the detection has not improved. The FX rates (Figures 4.2.15 and 4.2.16) show similar behavior, but since they do not suffer with such significant trend that would project onto the differentiated series, the Autoencoder detection indeed has improved. The comparison between Figures 4.2.15 and 4.2.2, as well as the Figures 4.2.16 and 4.2.3 shows that the anomaly detections are now not concentrated around single points (the 2007 and 2012 forEUR/JPY, 2022 for theEUR/USD), but are more distributed and better highlight signifficant shifts throughout the time series. For the commodities (Figures 4.2.17 and 4.2.18), the redistribution along the curve is also much more significant. We can see, especially on oil prices in Figures 4.2.18 and 4.2.5, that the points are not concentrated along the highest points, but rather on the most significant shifts. Gold, when comparing the figures 4.2.17 and 4.2.4 is also better reconstructed.

The Autoencoder results on the different series show similar behavior as for the original series. This is due to the trend that is best visible in Figures 4.2.13 and 4.2.14, where the trend is continuous and significant. This tells us that, overall, the Autoencoder works better on detrended data and that the differentiation itself does not help much.

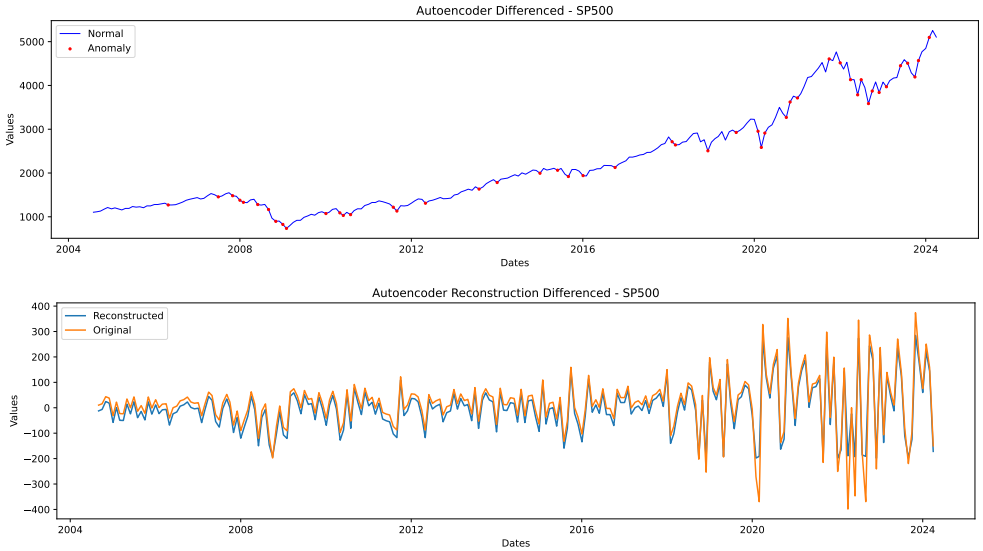

Figure 4.2.14 S&P500 index price anomalies and reconstruction for Autoencoder with differenced data.

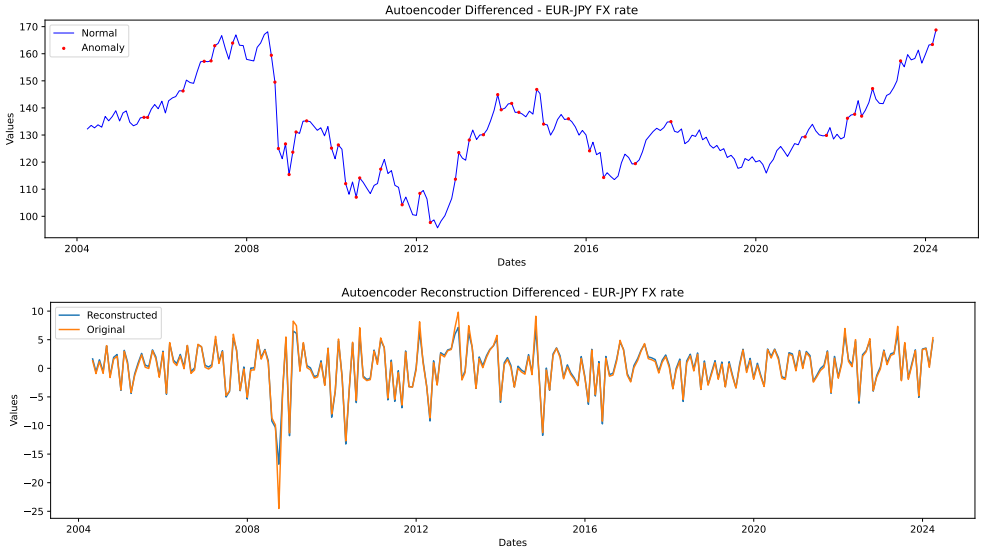

Figure 4.2.15 EUR/JPY FX rate anomalies and reconstruction for Autoencoder with differenced data.

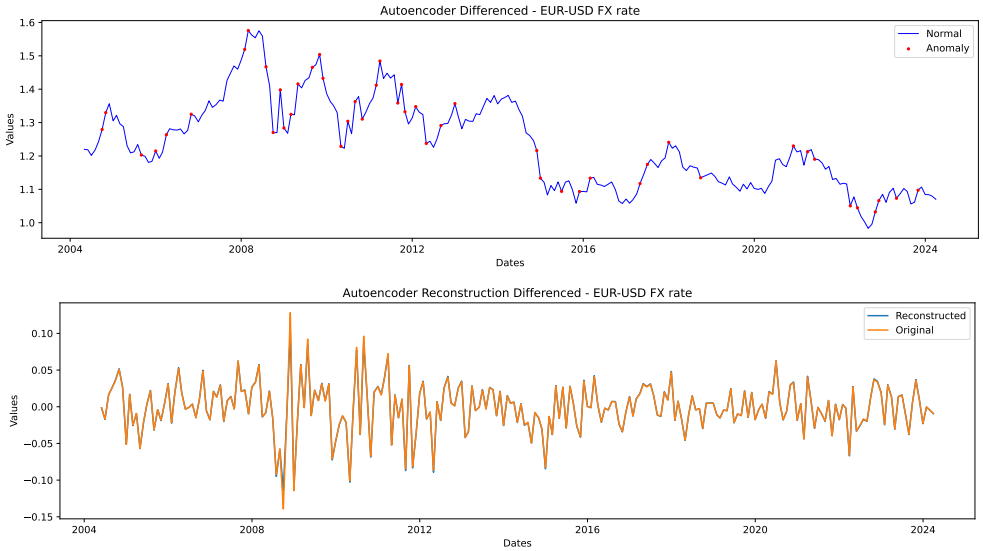

Figure 4.2.16 EUR/USD FX rate anomalies and reconstruction for Autoencoder with differenced data.

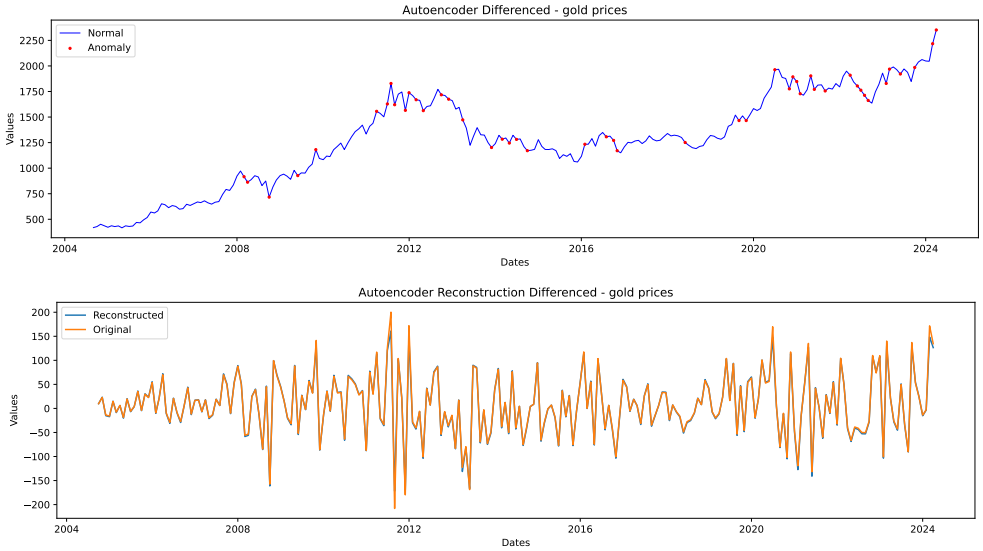

Figure 4.2.17 Gold price anomalies and reconstruction for Autoencoder with differenced data.

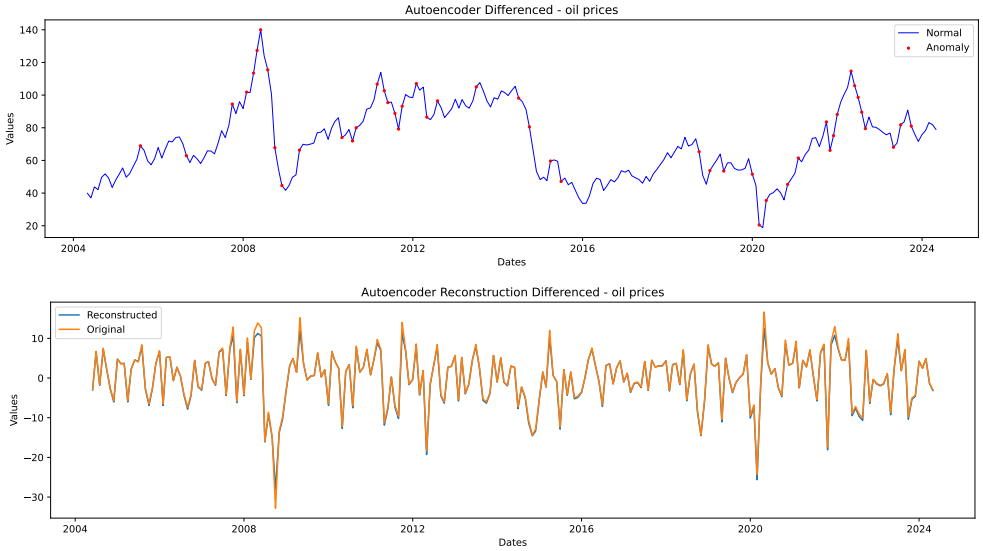

Figure 4.2.18 Oil price anomalies and reconstruction for Autoencoder with differenced data.

4.2.4 Data with relative change

The relative changes should give us better results since we eliminate the trends and seasonality.

As for the stock markets, we can see improvement; Figure 4.2.19 shows a better distribution of the anomalous dates, concentrated around the biggest market crashes (2008, 2016, and 2020). We may notice that contrary to the other graphs, the part after 2020 is much more sparsely labeled - this is because these changes are relatively small compared to the shifts during market crashes. S&P500 shows similar behavior; We can see that the distribution is similar for Figures 4.2.20 and 4.2.19.

The FX rates and commodities graphs (Figures 4.2.21, 4.2.22, 4.2.23 and 4.2.24) show similar improvement. We can see that the Autoencoder labels the anomalies more believably since the differences are now relative instead of absolute.

4.2.5 Conclusion

The Autoencoder performed very well on the data. Although not perfect, the reconstruction was great in the cases of differenced and relative change data. The usage of unadjusted or detrended data is not discouraged. However, the algorithm performs best on data without trends, with both reconstruction and subsequent anomaly detection being faithful to reality. The Autoencoder

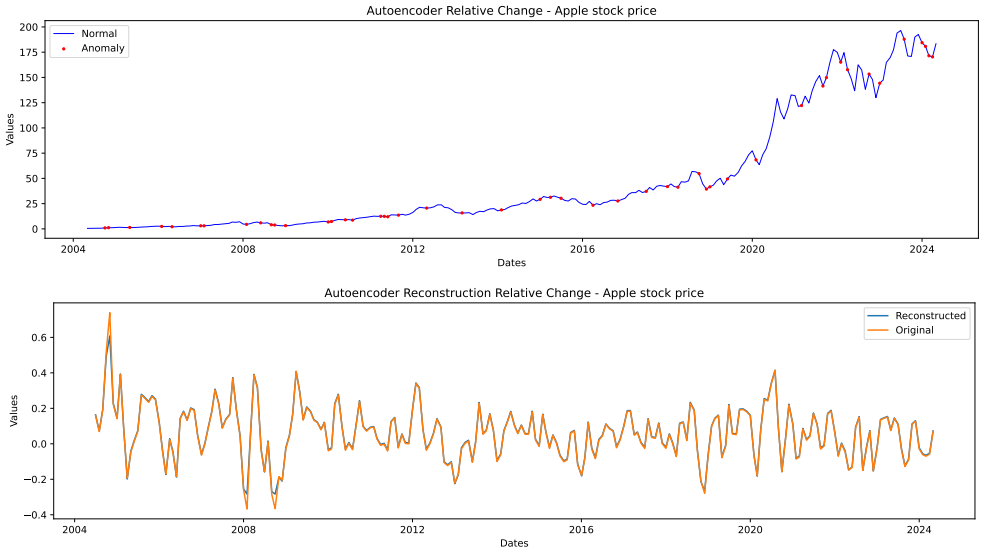

Figure 4.2.19 Apple stock price anomalies and reconstruction for Autoencoder with relative change time series.

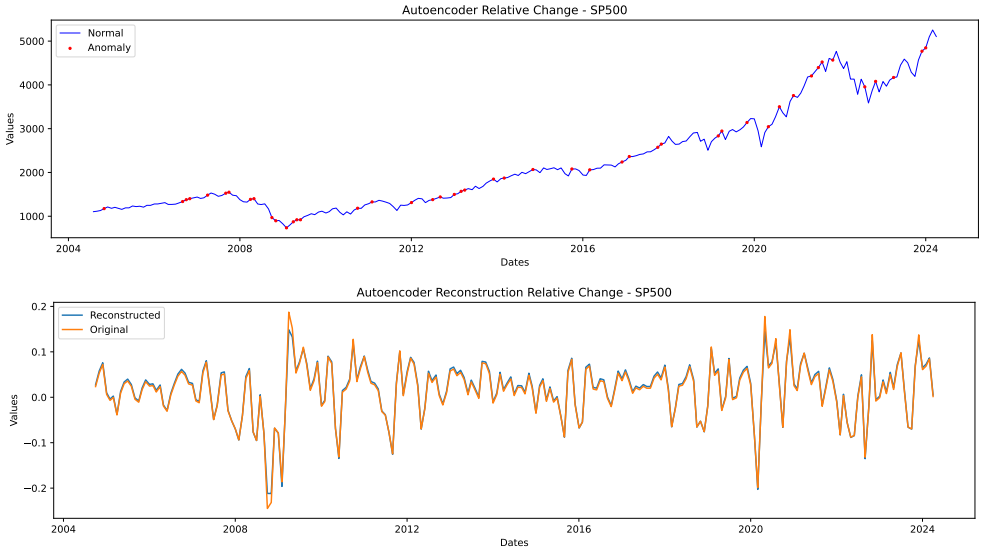

Figure 4.2.20 S&P500 index price anomalies and reconstruction for Autoencoder with relative change time series.

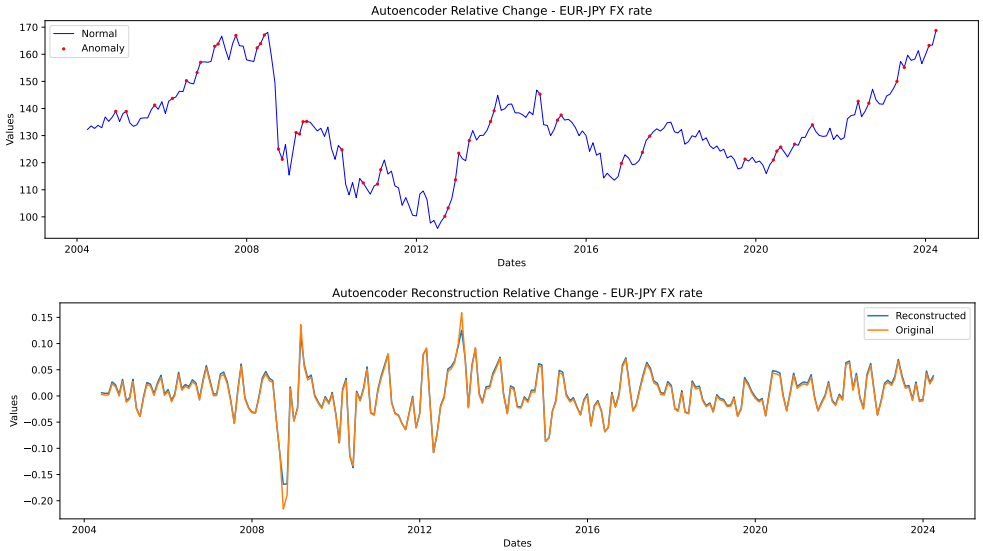

Figure 4.2.21 EUR/JPY FX rate anomalies and reconstruction for Autoencoder with relative change time series.

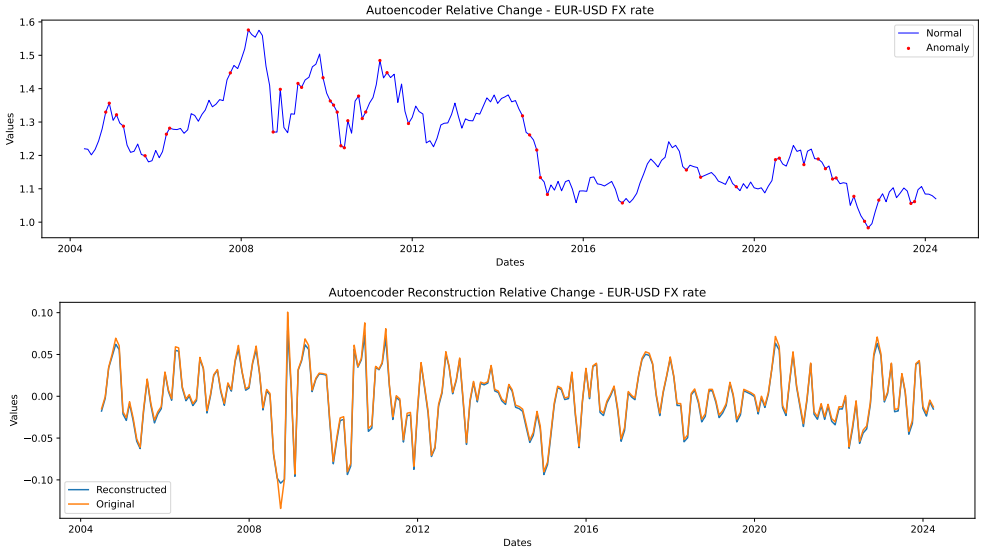

Figure 4.2.22 EUR/USD FX rate anomalies and reconstruction for Autoencoder with relative change time series.

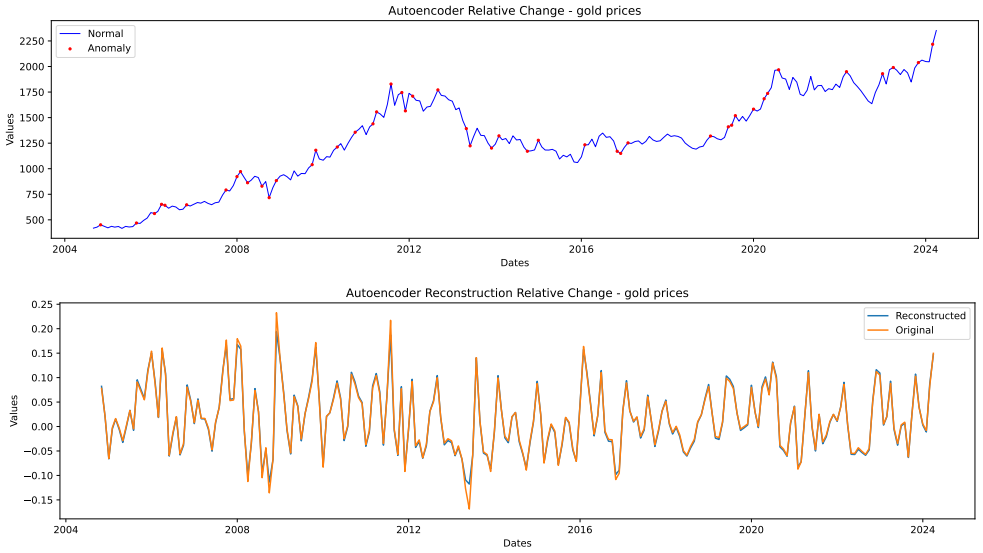

Figure 4.2.23 Gold price anomalies and reconstruction for Autoencoder with relative change time series.

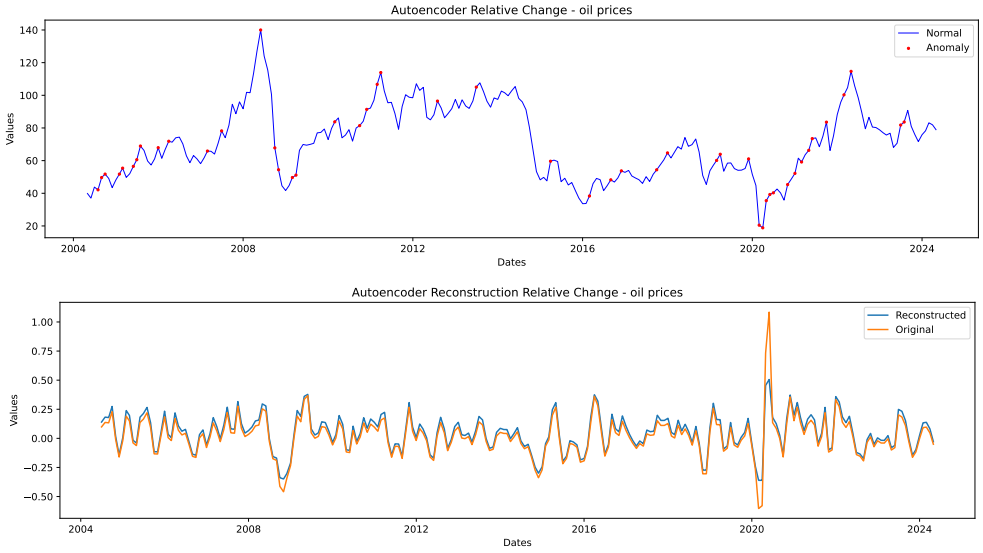

Figure 4.2.24 Oil price anomalies and reconstruction for Autoencoder with relative change time series.

models all of the asset classes very well, and the results correspond to the findings of the original research.

4.3 LSTM anomaly detection

4.3.1 Normal Data

For stocks, the LSTM model shows a good ability to reconstruct the patterns of the Apple stock price. In Figure 4.3.1, the LSTM's anomaly detection shows the advantages of its memory capability, in contrast with the Autoencoder's approach in Figure 4.2.1, which misses some subtle shifts seen by LSTM. The LSTM captures the main market dynamics, notably around significant dips and peaks that the Autoencoder often misses. However, we can see that the detections are still limited to the later dates. For the S&P500 index, shown in Figure 4.3.2, the LSTM model identifies anomalies with higher sensitivity compared to the Autoencoder model in Figure 4.2.6. The LSTM's use of past information helps it react to sudden changes better, unlike the Autoencoder which is limited by its immediate reconstruction ability. It also shows better results than for Apple's stocks. For FX rates, the LSTM demonstrates much better adaptation than the Autoencoder. The EUR-JPY rate anomalies and reconstruction in Figure 4.3.3 show that the LSTM captures anomalous points that the Autoencoder in Figure 4.2.2 misses. The LSTM's sequential nature makes it more suitable for predicting the flow of the FX rates, which leads to more accurate anomaly detection.

Similarly, the EUR-USD rate, depicted in Figure 4.3.4, benefits from LSTM's temporal understanding. The model highlights anomalies during significant shifts in the market, offering a better overview compared to the more static Autoencoder results in Figure 4.2.3. For commodities, the LSTM also performs well. For gold prices, shown in Figure 4.3.5, the LSTM model can detect anomalies in less volatile periods, and generally makes better detection than Autoencoder, which puts all anomalies in the initial and final part of the time series.

In the case of oil prices, in Figure 4.3.6, the LSTM again proves superior. The model reacts to the changes in the market, whereas the Autoencoder in Figure 4.2.5only looks at the highest peaks, missing details that the LSTM's memory cells pick up on.

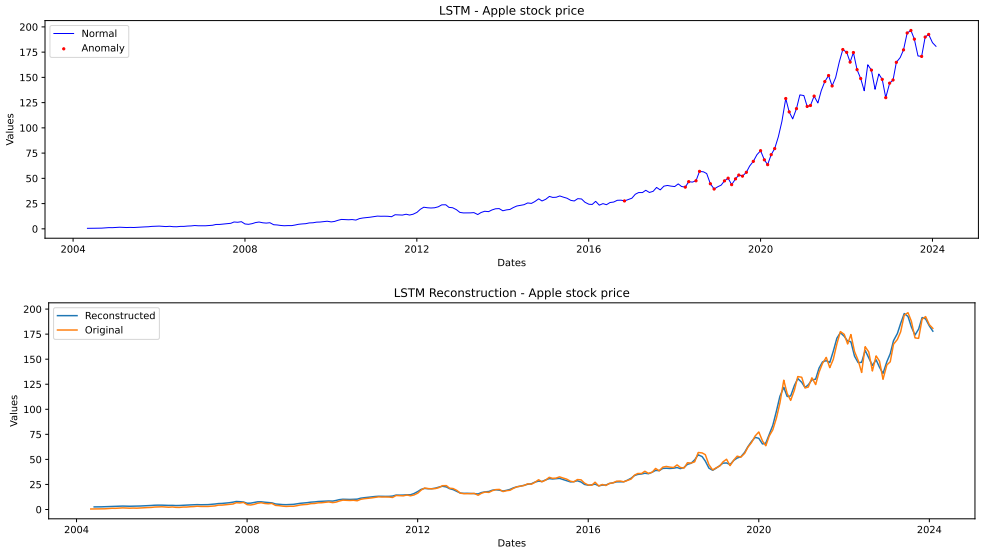

Figure 4.3.1 Apple stock price anomalies and reconstruction for LSTM.

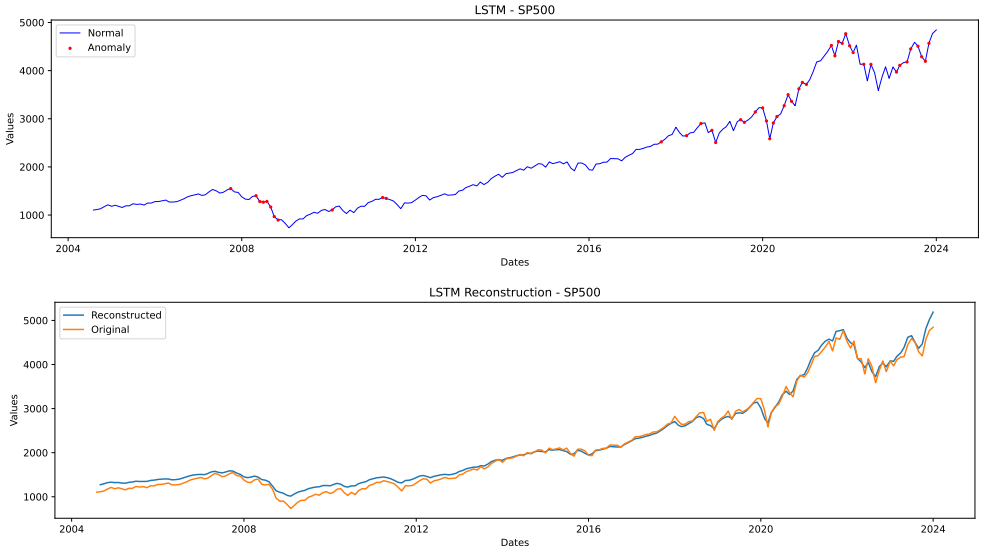

Figure 4.3.2 S&P500 index price anomalies and reconstruction for LSTM.

Figure 4.3.3 EUR/JPY FX rate anomalies and reconstruction for LSTM.

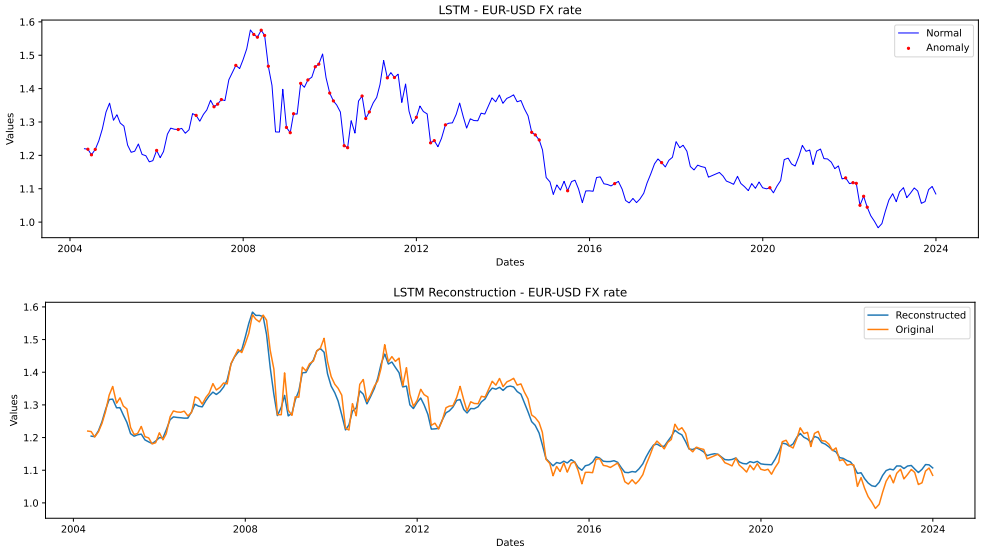

Figure 4.3.4 EUR/USD FX rate anomalies and reconstruction for LSTM.

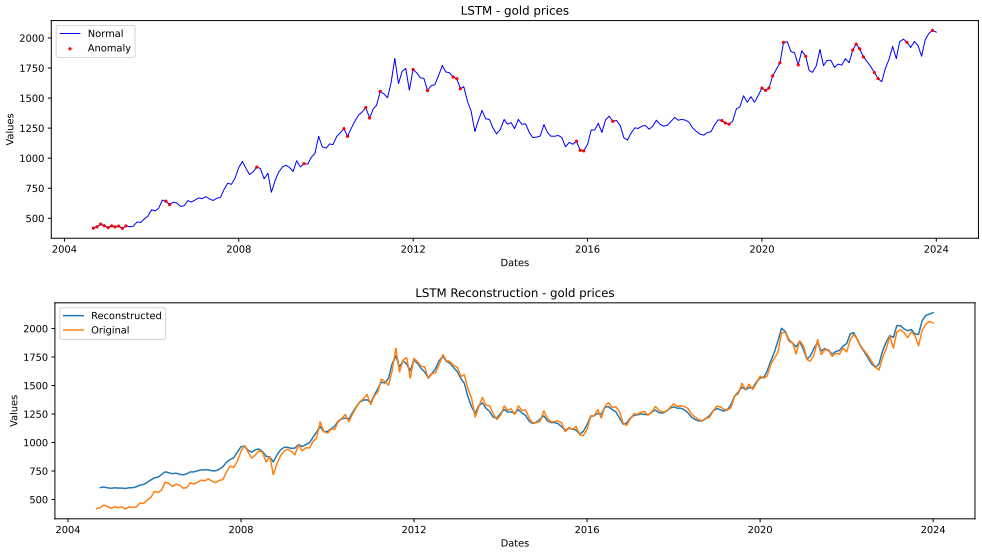

Figure 4.3.5 Gold price anomalies and reconstruction for LSTM.

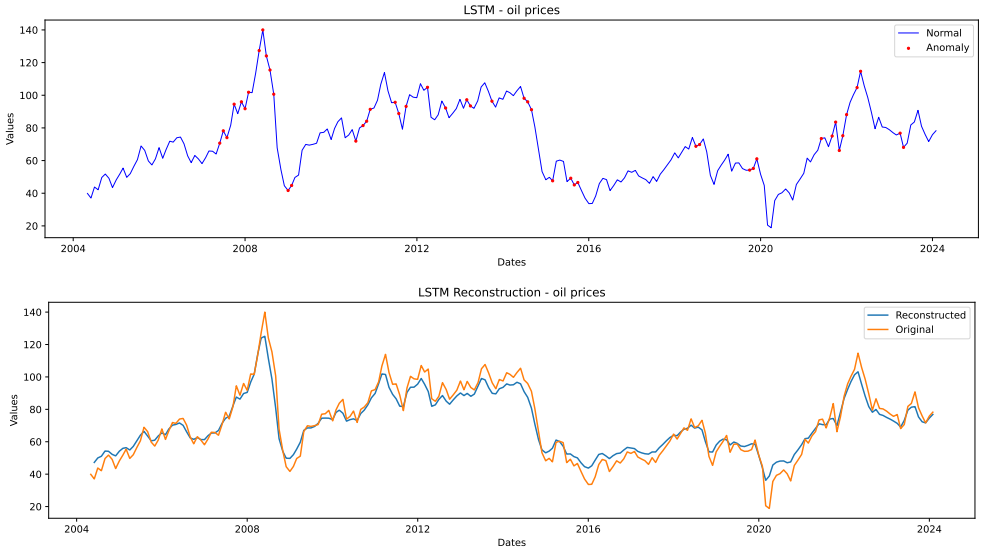

Figure 4.3.6 Oil price anomalies and reconstruction for LSTM.

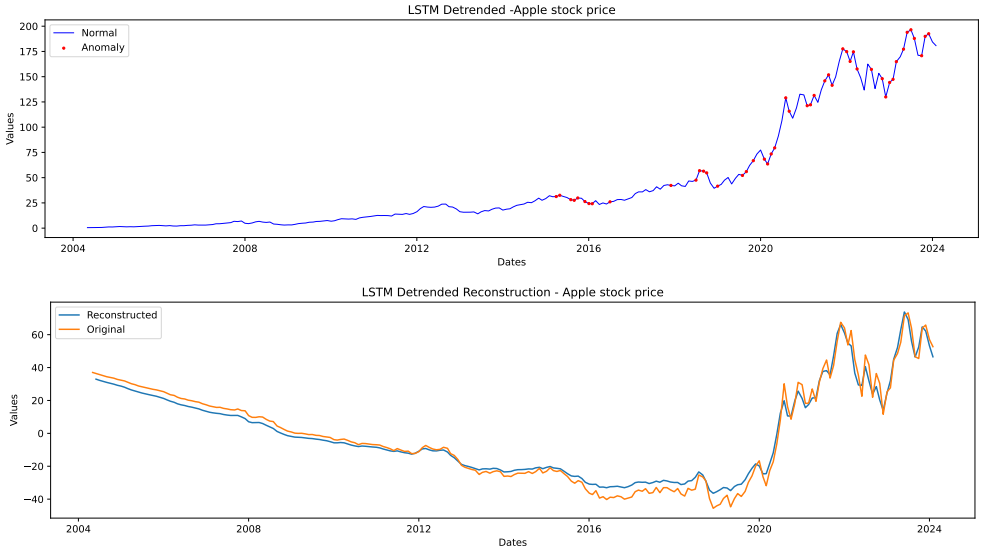

Figure 4.3.7 Apple stock price anomalies and reconstruction for LSTM with detrended data.

4.3.2 Detrended data

For detrended data, the LSTM model shows behavior similar to the one shown on the normal data. In the Apple stock price analysis using detrended data in Figure 4.3.7, the LSTM captures anomalies more consistently compared to the Autoencoder in Figure 4.2.7. However, it completely misses out on the first part of the graph. We can basically say that the absence of a trend does not damage or help the model. Figures 4.3.8, 4.3.9, 4.3.10, 4.3.12 and 4.3.11 confirm this. We can see that the reconstruction error is much smaller for the detrended data; however, the detection remains the same.

4.3.3 Differenced data

For the differenced data, we can see that the reconstruction does not go well; the fluctuations of the original series are much more significant than the reconstructed. However, the anomaly detection does not differ much from the original ones as we can see in Figures Figures 4.3.13, 4.3.14, 4.3.15, 4.3.16, 4.3.18 and 4.3.17. There are not many changes for the differenced data when compared to normal data in a Figures 4.3.1, 4.3.2, 4.3.3, 4.3.4, 4.3.6 and 4.3.5

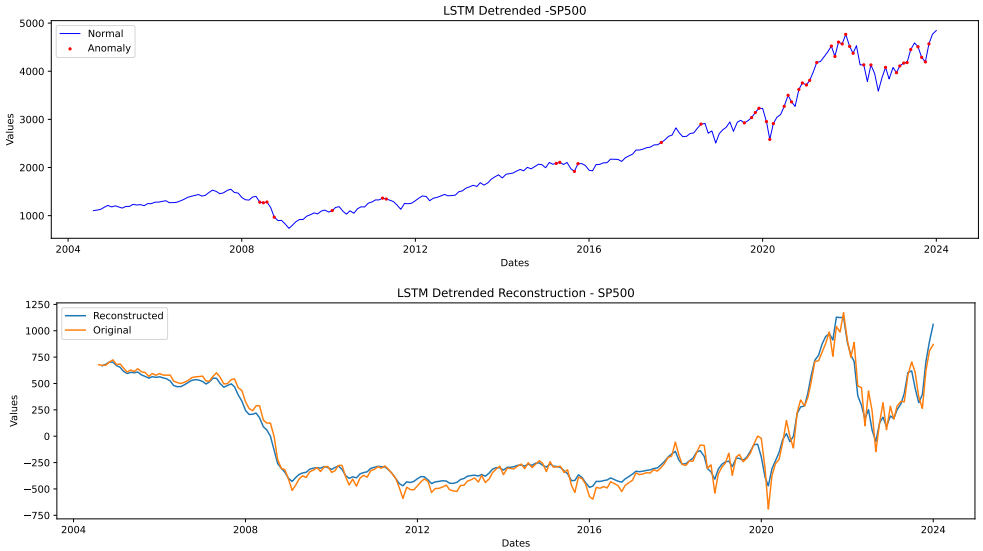

Figure 4.3.8 S&P500 index price anomalies and reconstruction for LSTM with detrended data.

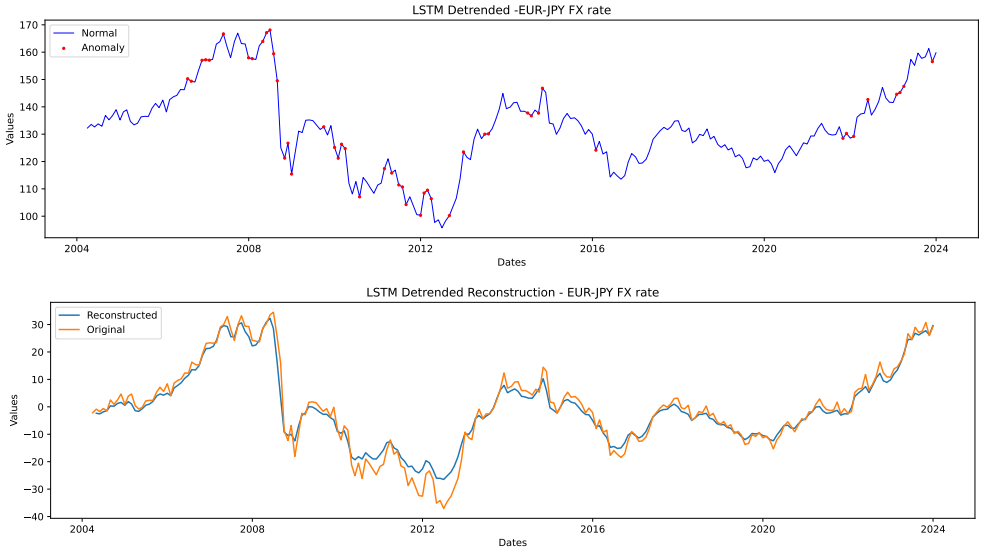

Figure 4.3.9 EUR/JPY FX rate anomalies and reconstruction for LSTM with detrended data.

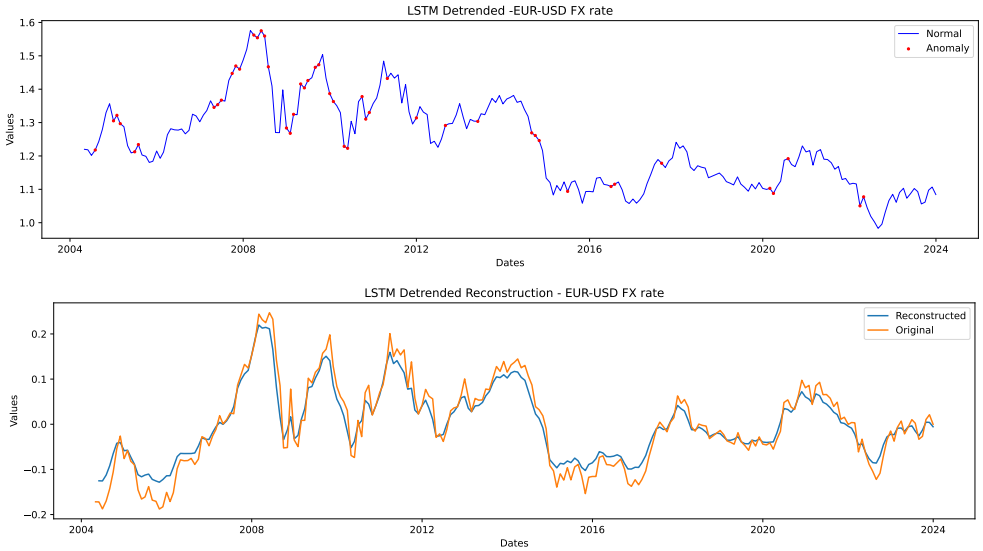

Figure 4.3.10 EUR/USD FX rate anomalies and reconstruction for LSTM with detrended data.

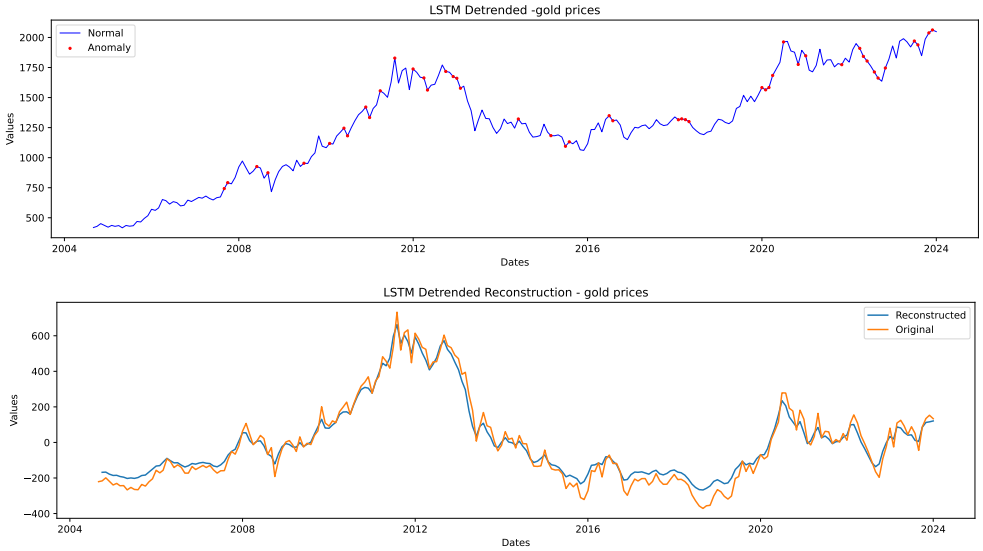

Figure 4.3.11 Gold price anomalies and reconstruction for LSTM with detrended data.

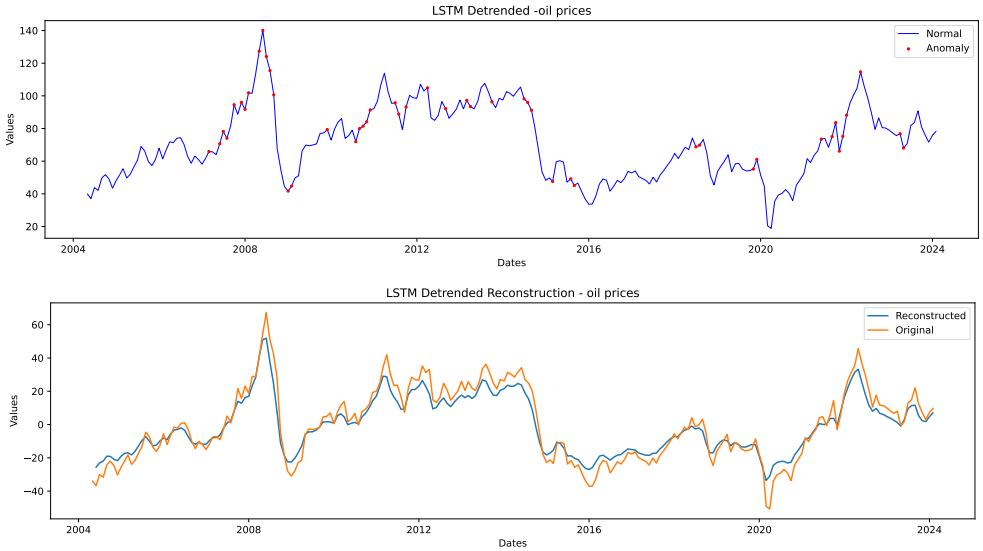

Figure 4.3.12 Oil price anomalies and reconstruction for LSTM with detrended data.

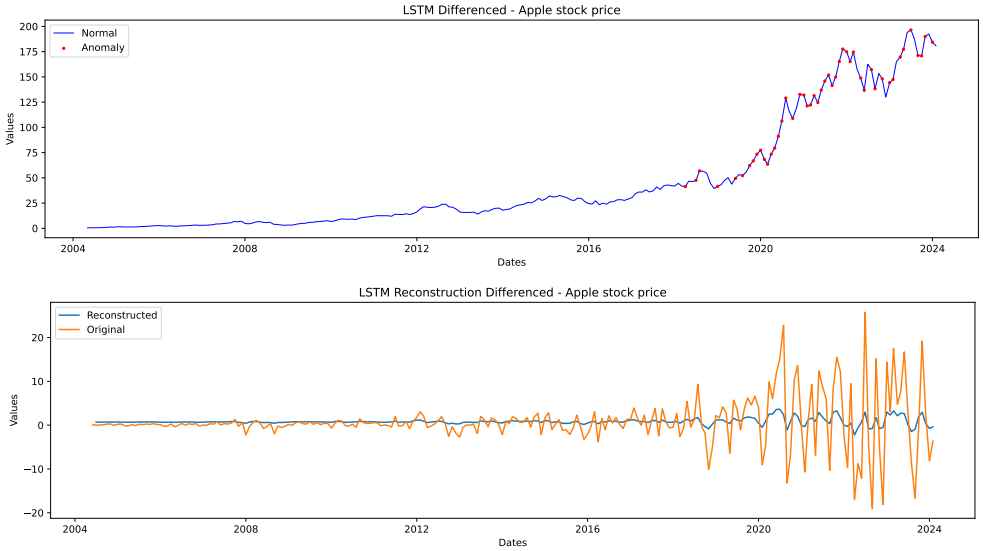

Figure 4.3.13 Apple stock price anomalies and reconstruction for LSTM with differenced data.

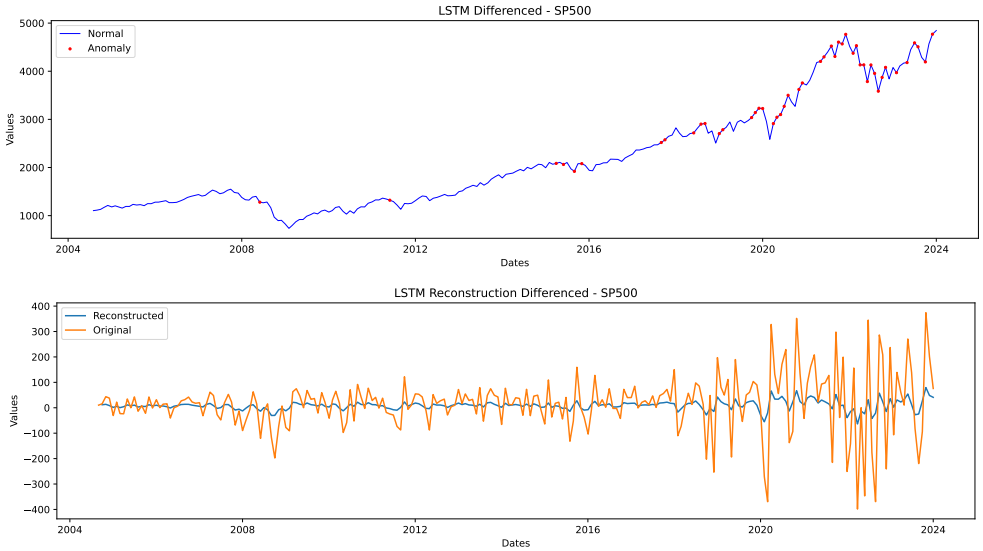

Figure 4.3.14 S&P500 index price anomalies and reconstruction for LSTM with differenced data.

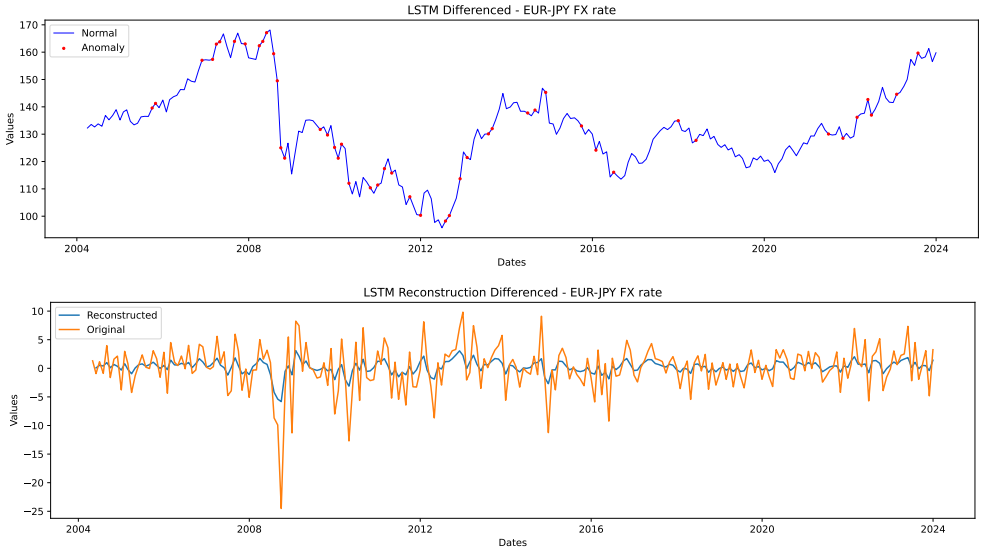

Figure 4.3.15 EUR/JPY FX rate anomalies and reconstruction for LSTM with differenced data.

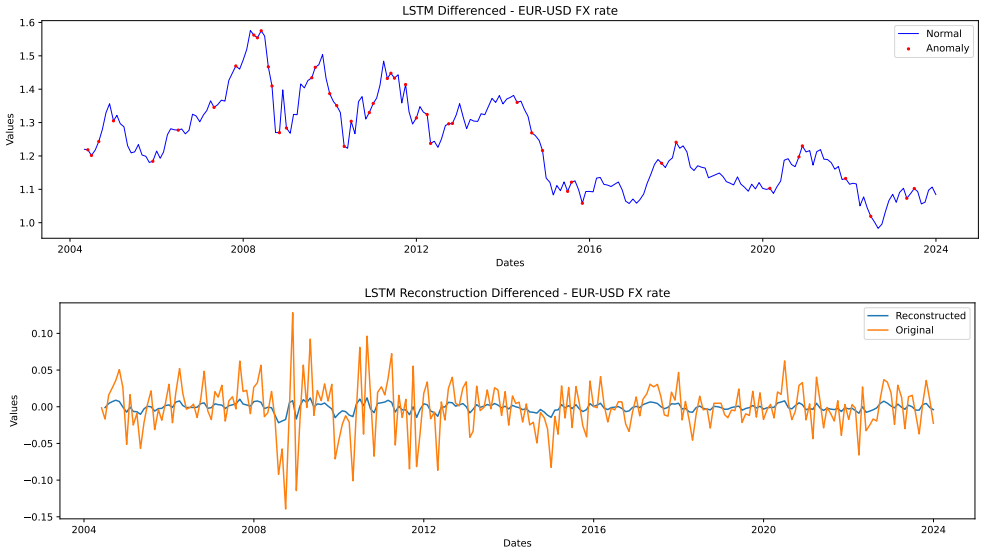

Figure 4.3.16 EUR/USD FX rate anomalies and reconstruction for LSTM with differenced data.

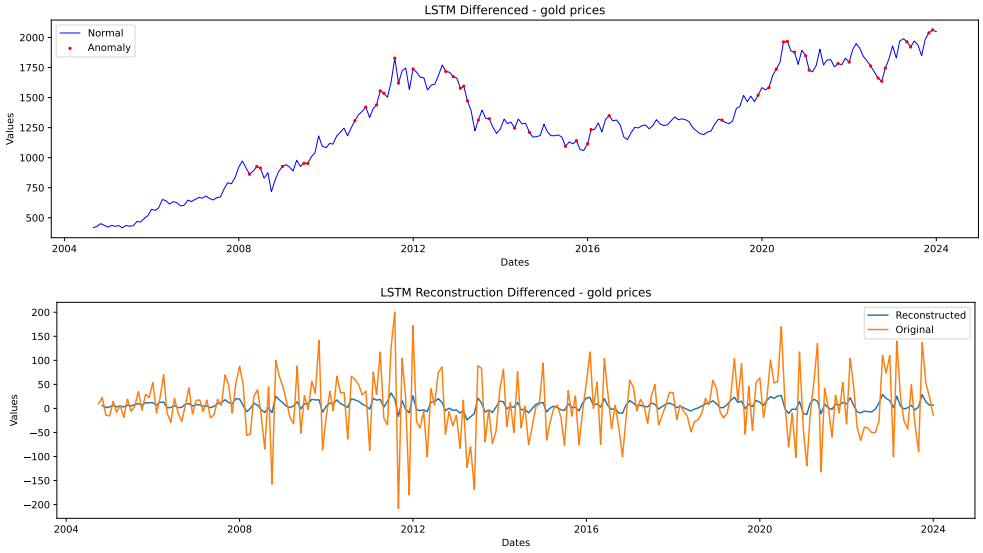

Figure 4.3.17 Gold price anomalies and reconstruction for LSTM with differenced data.

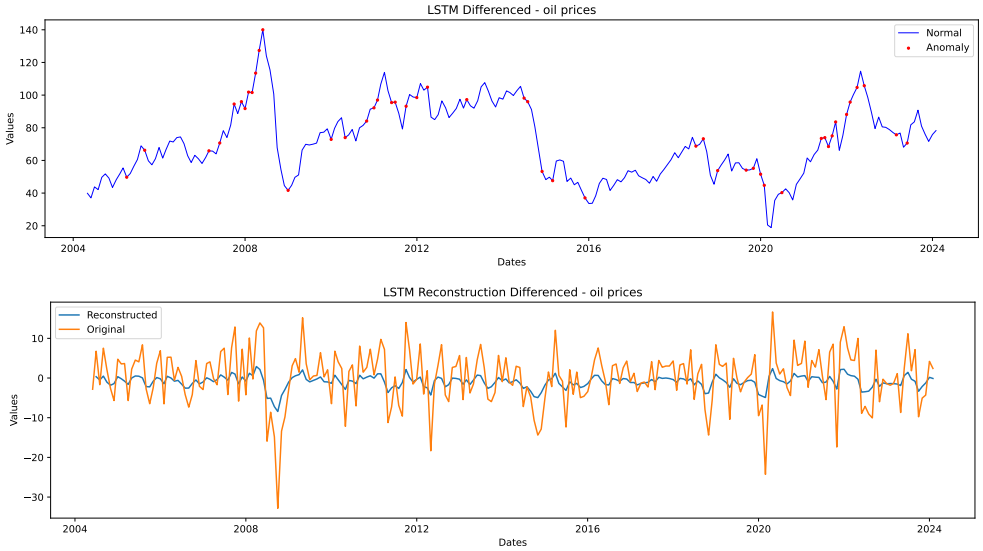

Figure 4.3.18 Oil price anomalies and reconstruction for LSTM with differenced data.

4.3.4 Data with relative change

For the relative change, we can see that the reconstruction is the same as it was for differenced data. However, we can see that the detections have changed.

The stocks (Figures 4.3.19 and 4.3.20) show many more places highlighted, where instead of concentrating around the biggest values, the points are now concentrated around the years where significant market shifts have happened.

The FX rates (Figures 4.3.21 and 4.3.22) do not show siginifficant improvement compared to the other methods.

For commodities, we can see minor improvement. The detection is good, especially the oil prices seem to be modeled better than in the case of Autoencoder in Figures 4.2.5, 4.2.11, 4.2.18 and 4.2.24, with lesser price adjustments market as anomalies and the anomalous points concentrating around more signifficant market shifts, however, this is also true for gold prices.

4.3.5 Conclusion

In conclusion, we can say that while not being particularly good at modeling stock markets, the LSTM showed good detection capabilities in commodities and FX rates. We saw that the reconstruction of the differenced data was not great. This may be because differencing transforms the data into a series of changes, which can be harder for the LSTM to model due to the loss of absolute scale and trend information. LSTMs often rely on the continuity

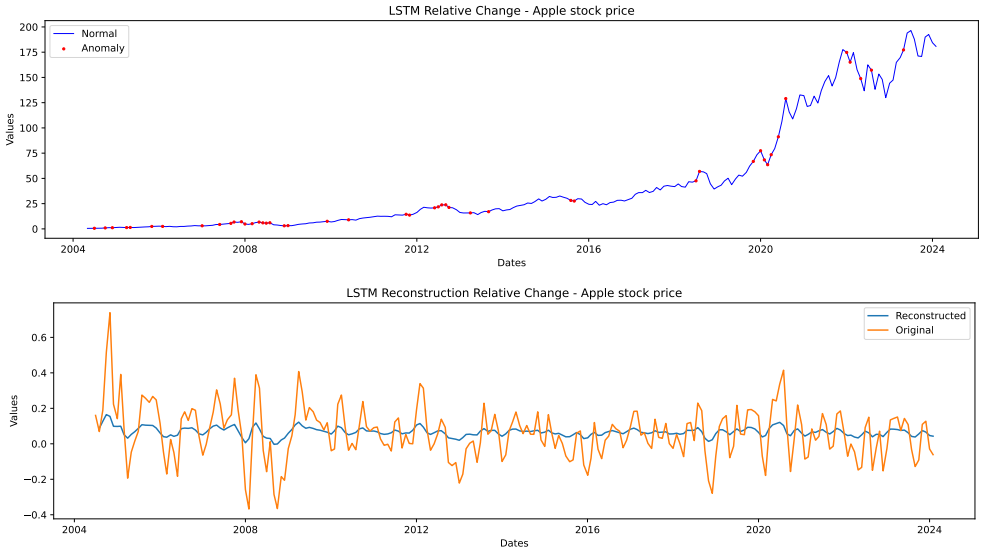

Figure 4.3.19 Apple stock price anomalies and reconstruction for LSTM with relative change time series.

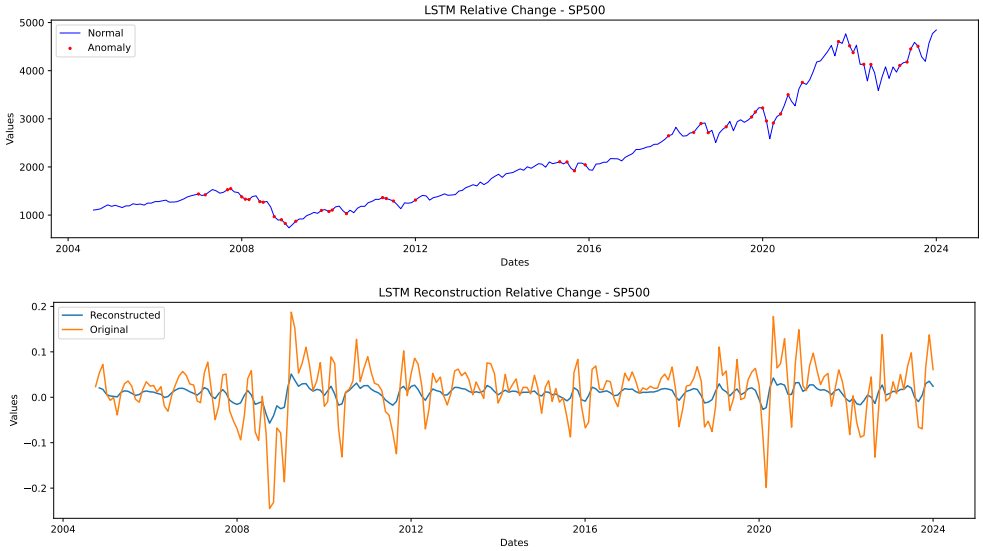

Figure 4.3.20 S&P500 index price anomalies and reconstruction for LSTM with relative change time series.

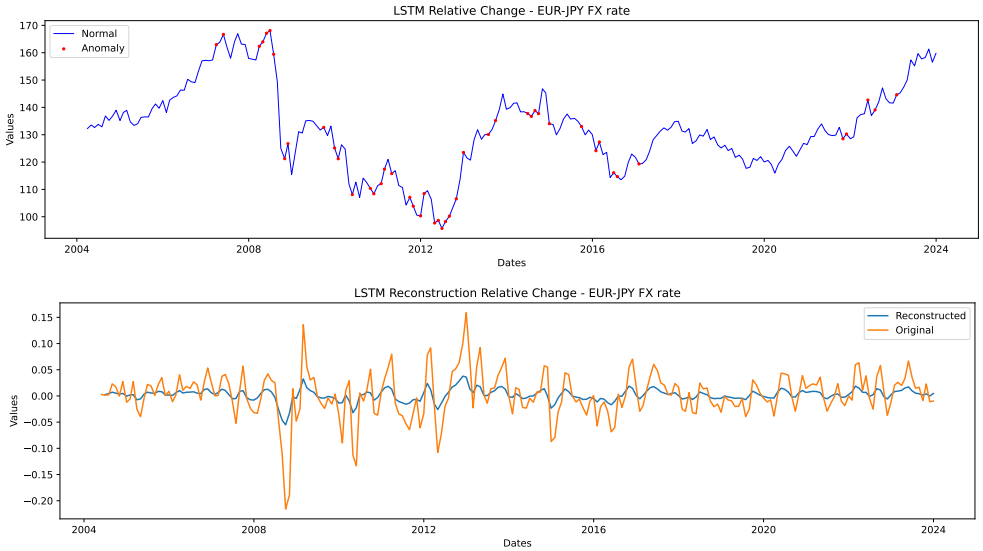

Figure 4.3.21 EUR/JPY FX rate anomalies and reconstruction for LSTM with relative change time series.

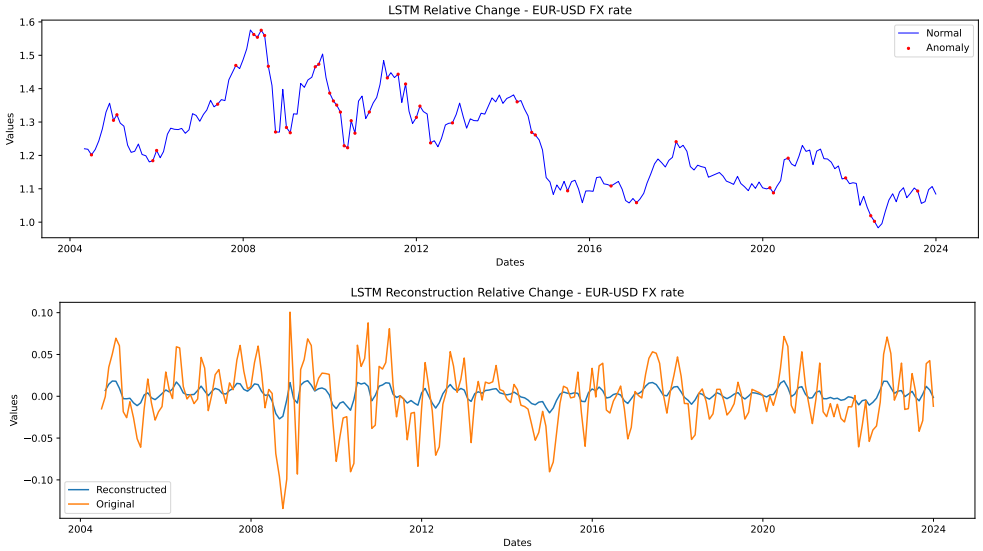

Figure 4.3.22 EUR/USD FX rate anomalies and reconstruction for LSTM with relative change time series.

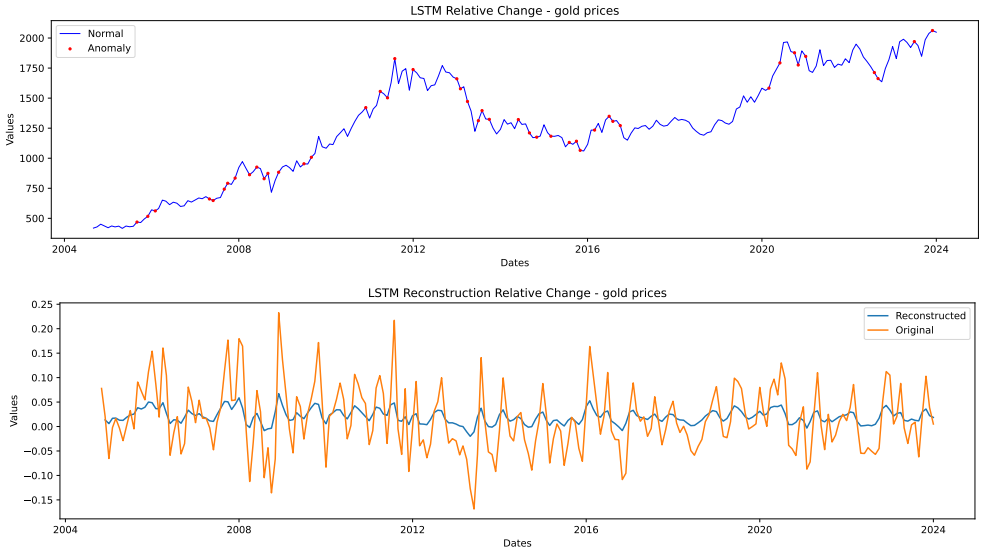

Figure 4.3.23 Gold price anomalies and reconstruction for LSTM with relative change time series.

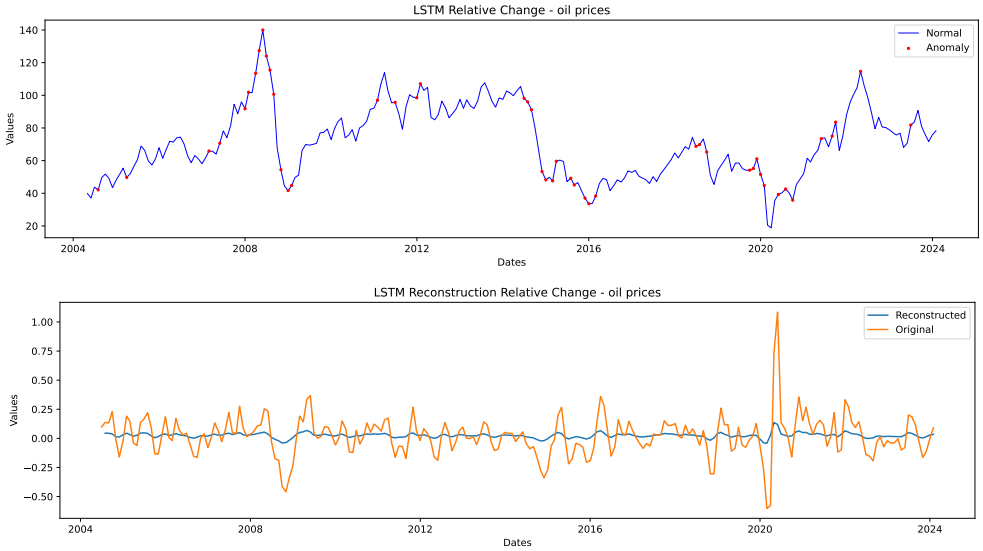

Figure 4.3.24 Oil price anomalies and reconstruction for LSTM with relative change time series.

and persistence of data to learn effective representations. When the data is differenced, the resulting values represent changes rather than actual levels, which can disrupt the LSTM's ability to learn the long-term dependencies effectively.

4.4 Hurst Exponent anomaly detection

4.4.1 Normal Data

Since there is no reconstruction for the Hurst exponent, we will just describe the quality of the detection. The stock market detection in Figures 4.4.1 and 4.4.2 is different from what we have seen in Figures 4.2.1 and 4.3.1 describing Apple stock modeling with Autoencoder and LSTM respectively, or Figures 4.2.6 and 4.3.2 describing the same for S&P500. We can see that the predicted anomalies are clustered in clusters throughout the time series and not concentrated on the last four years. However, the algorithm seems to be focusing on macroscopic market efficiency trends rather than looking for anomalies. For the FX rates (Figures 4.4.3 and 4.4.10), we can again see a focus on less important trends in the data, while missing out significant market shifts.

4.4.2 Detrended Data

For detrended data, the anomaly detections were very similar to the non-adjusted dataset. This can be explained by the fact that the Hurst Exponent primarily measures the long-term memory of a time series, which is not significantly altered by simply removing a linear trend. The persistence characteristics that Hurst identifies — whether a series tends to revert to its mean (anti-persistent behavior) or continue in its current direction (persistent behavior) — are largely invariant under linear transformations like detrending.

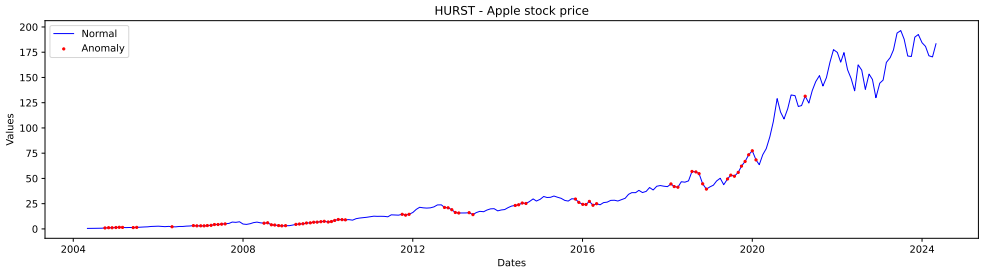

Figure 4.4.1 Apple stock price anomalies Hurst Exponent.

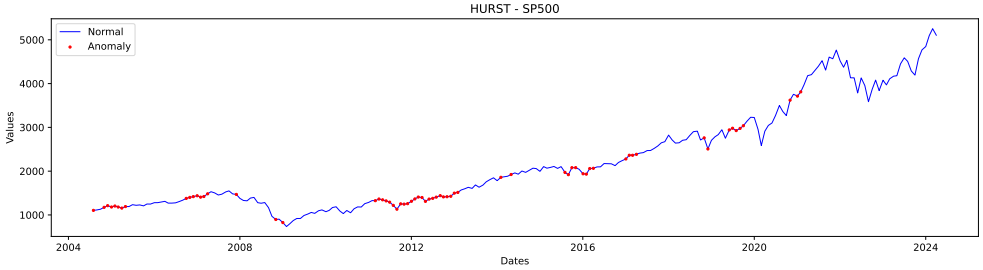

Figure 4.4.2 S&P500 index price anomalies Hurst Exponent.

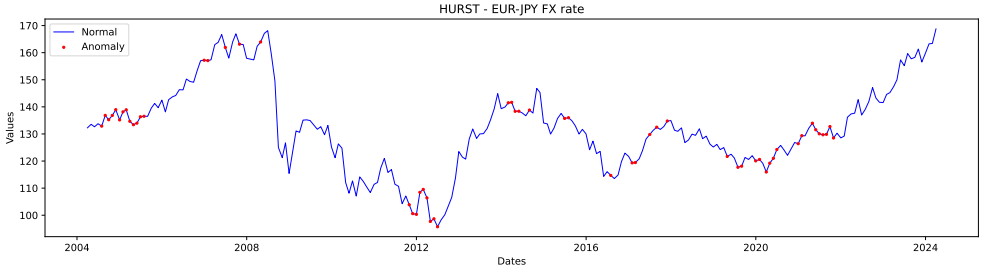

Figure 4.4.3 EUR/JPY FX rate anomalies Hurst Exponent.

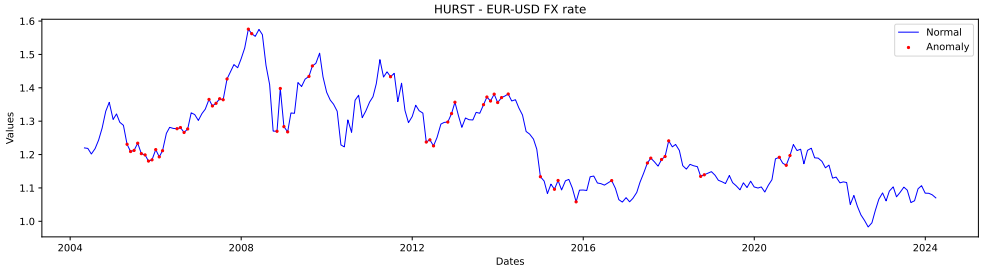

Figure 4.4.4 EUR/USD FX rate anomalies Hurst Exponent.

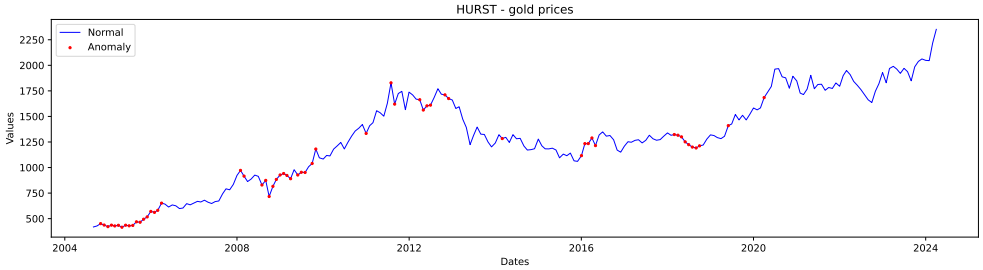

Figure 4.4.5 Gold price anomalies Hurst Exponent.

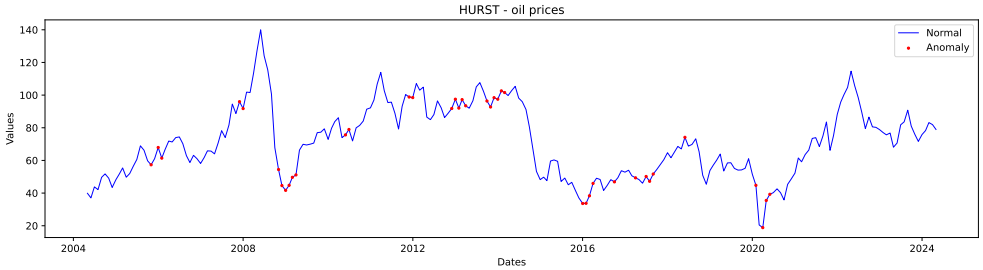

Figure 4.4.6 Oil price anomalies Hurst Exponent.

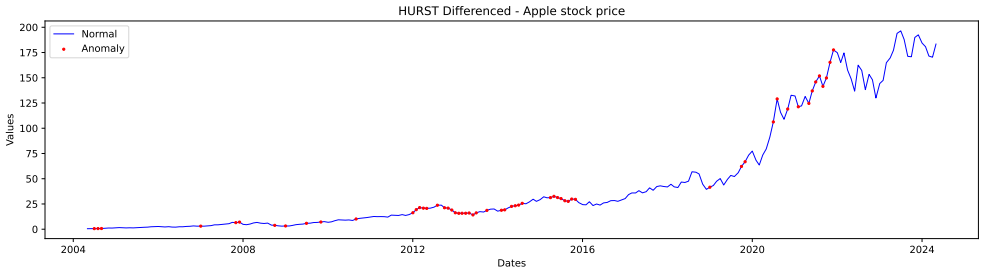

Figure 4.4.7 Apple stock price anomalies Hurst Exponent with differenced data.

4.4.3 Differenced Data

Contrary to the detrended data, the differenced dataset shows slight differences in modeling the anomalies.

For the Apple stocks depicted in Figure 4.4.7, we can see that the anomalies are less concentrated around the year 2008, and the model highlights other years as anomalies as well. For the S&P500, the improvement is similar, as shown in Figure 4.4.8. Still, the perceived anomalies are clustered together in both cases, even though all of the major market capitalization shifts are covered.

For the FX rates, we can see similar behavior as to the one in Figures

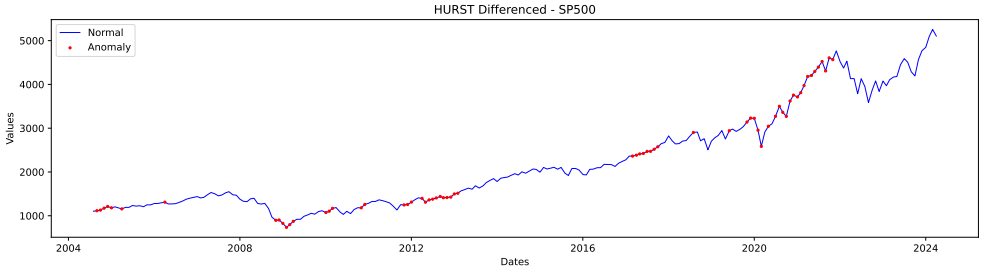

Figure 4.4.8 S&P500 index price anomalies Hurst Exponent with differenced data.

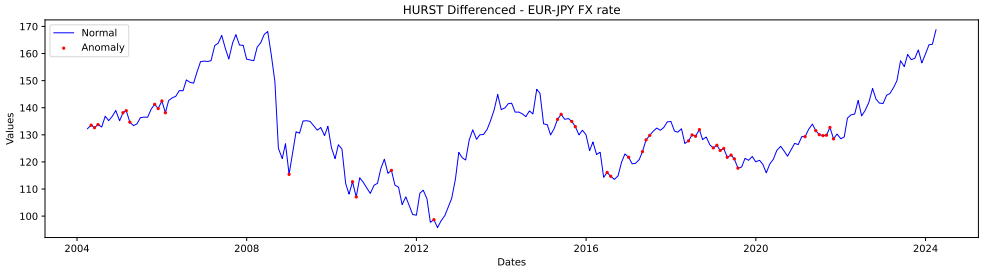

Figure 4.4.9 EUR/JPY FX rate anomalies Hurst Exponent with differenced data.

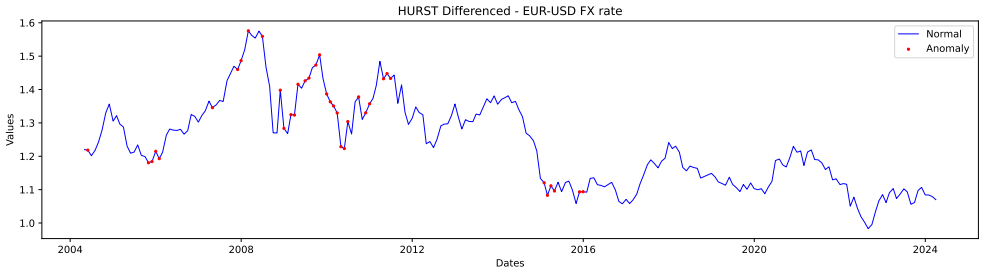

Figure 4.4.10 EUR/USD FX rate anomalies Hurst Exponent with differenced data.

4.4.3 and 4.4.4. For the EUR/JPY FX rate in Figure 4.4.9, there is a slight deterioration in the anomaly detection quality - we can see that the algorithm starts concentrating on small shifts before the year 2020 and leaves out the big sudden changes. For the EUR/USD graph in Figure 4.4.10, the anomaly points are concentrated around the year 2008, and the other important shifts are ignored.

As for the commodities, the results were ambiguous. The oil price curve in Figure 4.4.12 shows interesting behavior with all the points concentrated around the beginning, completely ignoring almost all other big shifts, and the points outside of the beginning labeled as anomalies are usually just minor

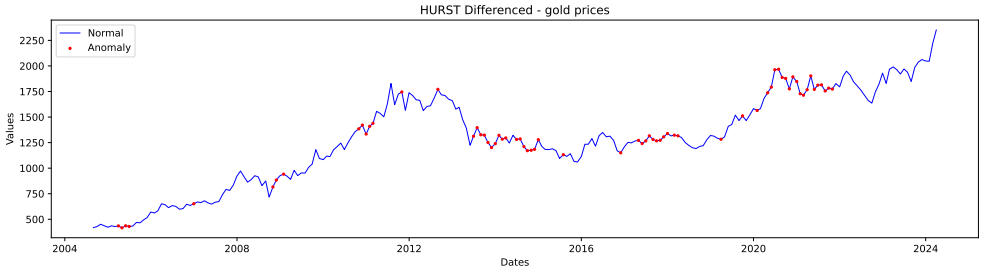

Figure 4.4.11 Gold price anomalies Hurst Exponent with differenced data.

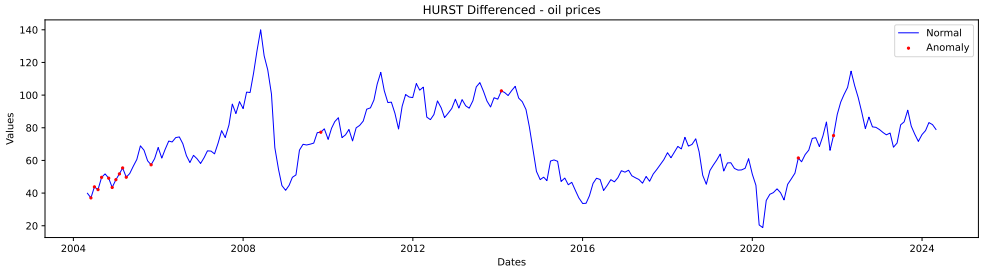

Figure 4.4.12 Oil price anomalies Hurst Exponent with differenced data.

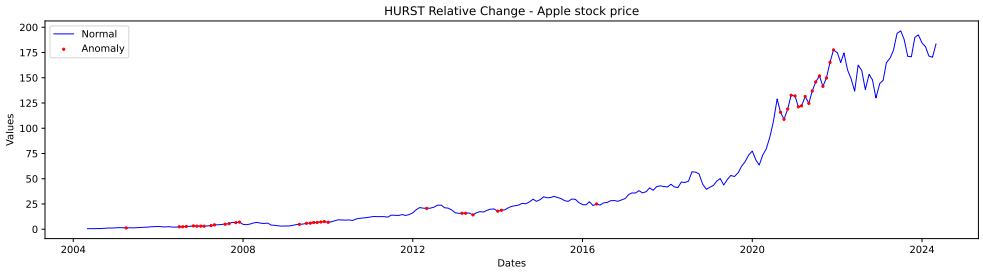

Figure 4.4.13 Apple stock price anomalies Hurst Exponent with relative change time series.

price adjustments, rather than indicators of a significant market shift. The gold price curve in Figure 4.4.11, on the other hand, turns out to be the other way around - the anomaly labels that were concentrated at the beginning in Figure 4.4.5 are now distributed in some other places. For this graph, the Hurst exponent worked very well.

4.4.4 Data with relative change

For the relative change time series, we can see that the Apple stocks in Figure 4.4.13 seem like a crossover between the Hurst exponent used on normal data in Figure 4.4.1 and differenced data in Figure 4.4.7. It does not resemble the other two algorithms when used with this time series. This can be due to the Hurst Exponent adapting to long-time correlations, which is shown in the first part of the graph, and then being unable to react accordingly during stronger shifts at the end of the time series. For the S&P500 depicted in Figure 4.4.14, the number of highlighted points is lower - this means that more samples here seem to be stable. However, again, the samples labeled as anomalies are small price adjustments rather than real market shifts.

For the FX rates in Figures 4.4.16 and 4.4.15, we can see that for both graphs, the number of anomalies is different than for differenced data (see

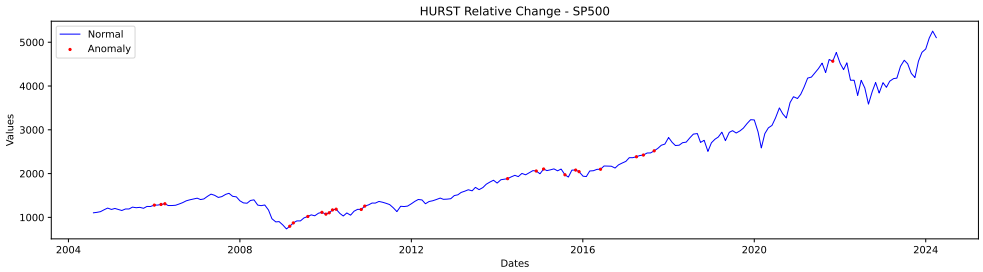

Figure 4.4.14 S&P500 index price anomalies Hurst Exponent with relative change time series.

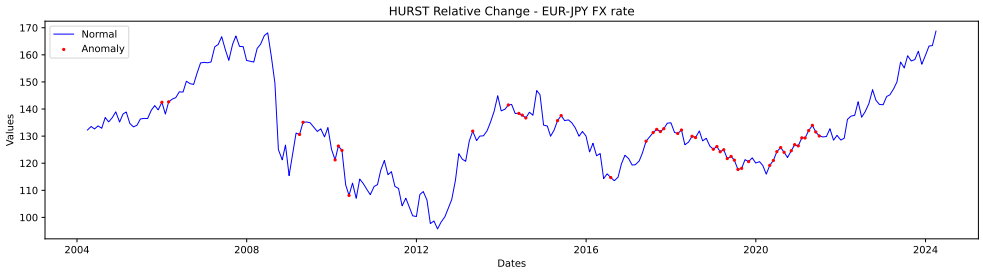

Figure 4.4.15 EUR/JPY FX rate anomalies Hurst Exponent with relative change time series.

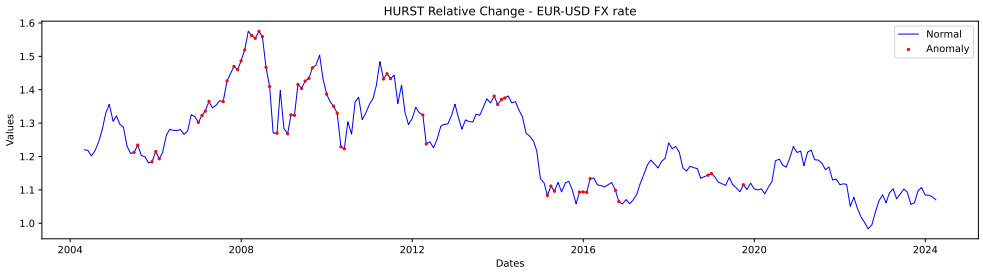

Figure 4.4.16 EUR/USD FX rate anomalies Hurst Exponent with relative change time series.

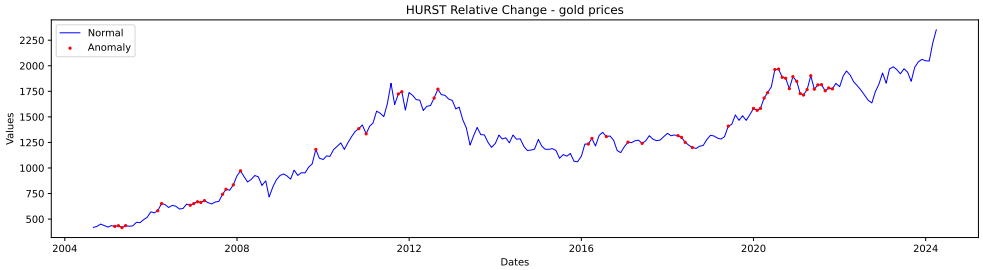

Figure 4.4.17 Gold price anomalies Hurst Exponent with relative change time series.

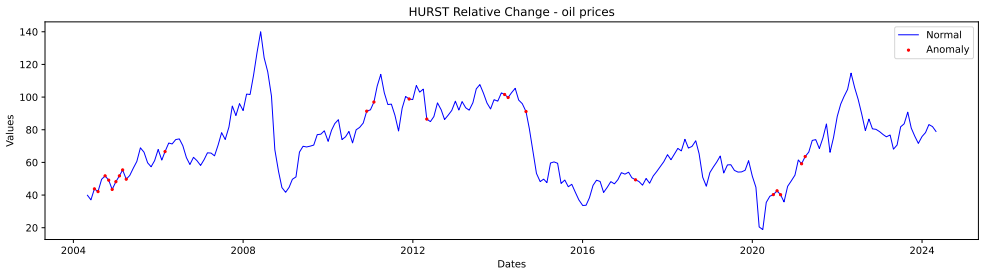

Figure 4.4.18 Oil price anomalies Hurst Exponent with relative change time series.

Figures 4.4.10 and 4.4.9). Interestingly, for the EUR/USD, there are more anomaly labels that are placed in the correct places - around the years 2008-2011, as well as 2015. For the second one, on the contrary, we can see that the number of labels is smaller, although the ones that persisted are in the same spots as in the differenced series. Overall, the improvement is only visible in the EUR/USD graph.

As for commodities, we can see from Figure 4.4.18 that the number of labels is still low, the same case as it was for differenced data in Figure 4.4.12. This again suggests that the algorithm couldn't adapt to the time series and thus found little to no anomalies. The gold curve in Figure 4.4.17 also shows much fewer anomalies than in the previous two cases (Figures 4.4.5 and 4.4.11).

4.4.5 Conclusion

As we can see, the Hurst Exponent did not perform particularly well in modeling anomalies for stock market data, particularly for the Apple stock price and the S&P 500 index. The detected anomalies were usually microscopic shifts, and the large anomalies were left unnoticed. Although there were minor improvements in anomaly detection with differenced data for Apple stock and S&P 500, overall, the Hurst Exponent did not adapt well to any dataset.

There was inconsistency in performance across different types of data, although the performance was not good. This suggests that this method is more sensitive to long-term, persistent behaviors in the data rather than short-term, sharp anomalies and, therefore, not a good fit for the purposes of our research.

4.5 Isolation Forests anomaly detection

4.5.1 Normal Data

For the Apple stocks in Figure 4.5.1, we can see that the output is somewhat similar to the output of the LSTM in Figure 4.3.1. Where only the samples after the year 2020 are highlighted as anomalies. If we look at the S&P500 in Figure 4.5.2, we can see that the algorithm works better than the previous ones (see Figures 4.2.6, 4.3.2 and 4.4.2) by also highlighting the year 2008 besides the anomalies after the year 2020, however, it still misses lots of anomalies, particularly anomalous samples in 2012, 2016 and 2018.

For the FX rates depicted in Figures 4.5.3 and 4.5.4, the algorithm shows similar behavior as for the same series processed by Autoencoder (see Figures 4.2.2 and 4.2.3). We can see that it searches for anomalies mainly in the peaks and dips in the graph and does not notice anomalies in 2005 for the/USD curve and 2016 for the/JPY curve. As for commodities, we can see from Figures 4.5.5 and 4.5.6 that the for the oil curve algorithm repeats the same pattern as for the FX rate, however for the gold prices, it does a better job by labeling correctly non-extreme data, like the 2016 drop.

4.5.2 Detrended Data

As we have seen in the previous section, the algorithm suffers from the presence of a trend like all of the other algorithms. In this section, we will have a look at how the detrending helps the algorithm to correctly identify outliers.

As for the Apple stocks (see Figure 4.5.7), we can see that detrending the data indeed had positive effects on the anomaly detection, which is also confirmed by the isolation score graph. We can see that the algorithm does not highlight just the most significant values but also the values around the years 2008 and 2013, where, in the original graph, we can see slight shifts that, at the time, were considered significant, especially for a company with such robust and stable market capitalization as Apple.

For S&P500 (Figure 4.5.8), the Isolation Forest does a good job identifying nearly all bigger shifts and market crashes. We can see on the isolation graph that the detrending here was better than in the case of the Apple stock graph

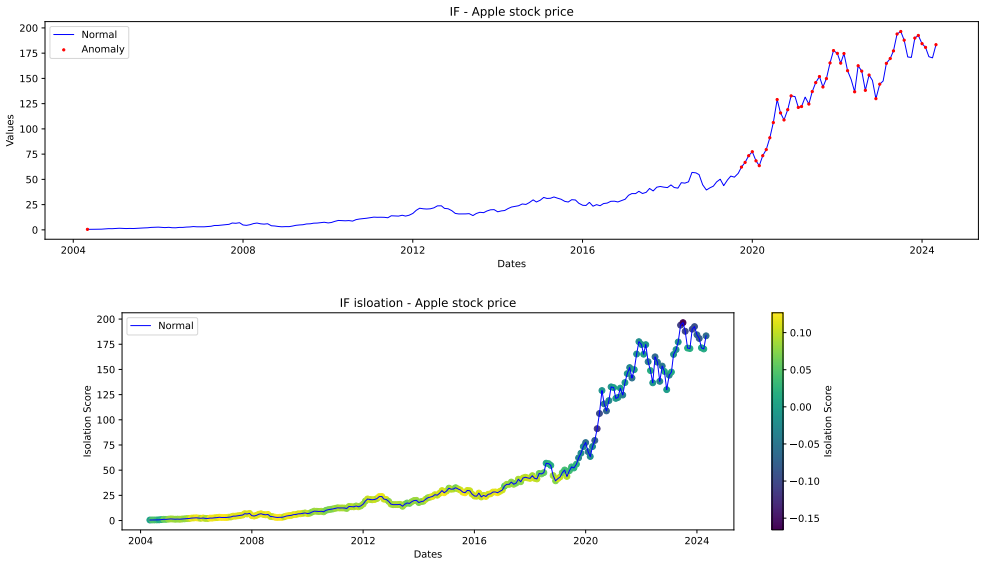

Figure 4.5.1 Apple stock price anomalies and separation score for Isolation Forest.

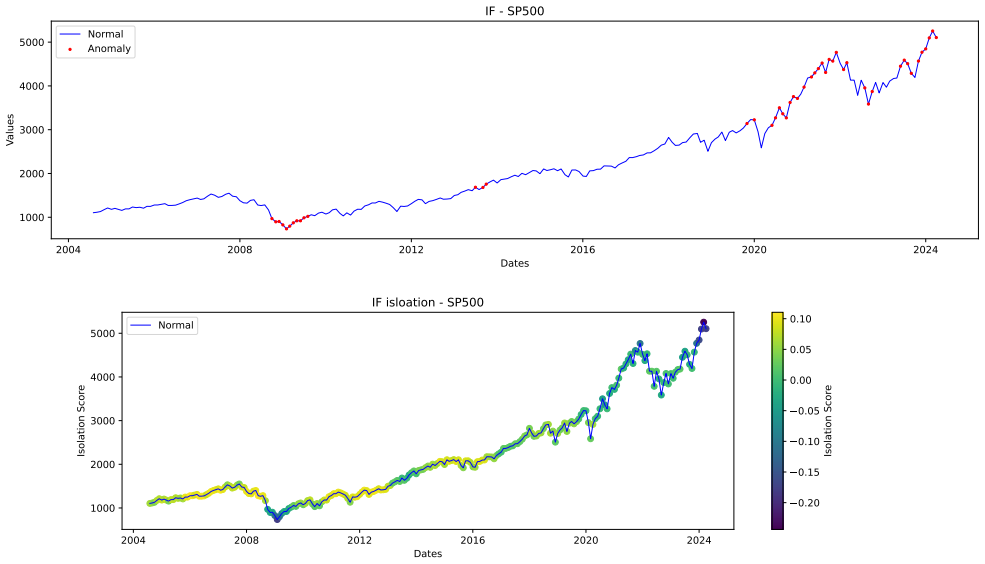

Figure 4.5.2 S&P500 index price anomalies and separation score for Isolation Forest.

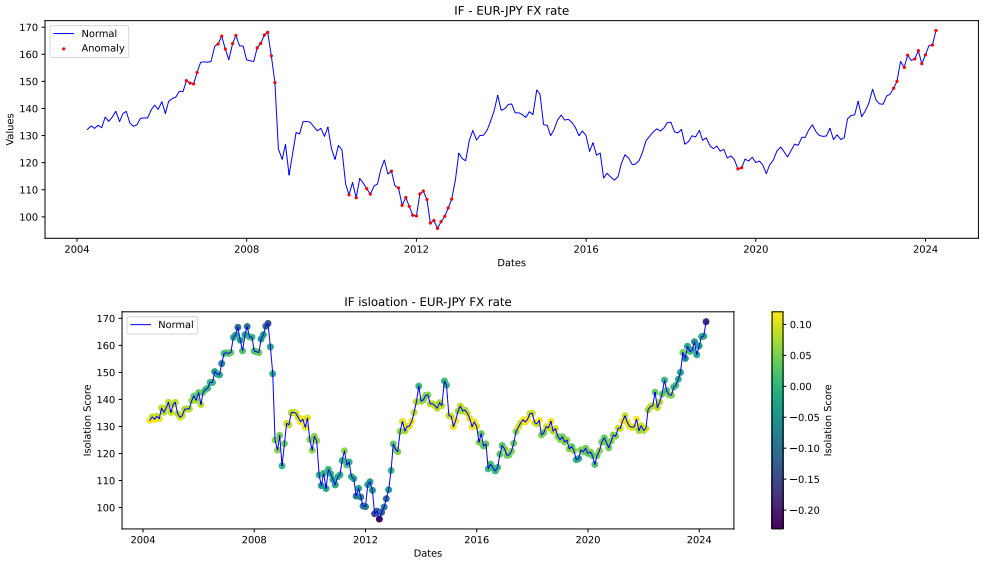

Figure 4.5.3 EUR/JPY FX rate anomalies and separation score for Isolation Forest.

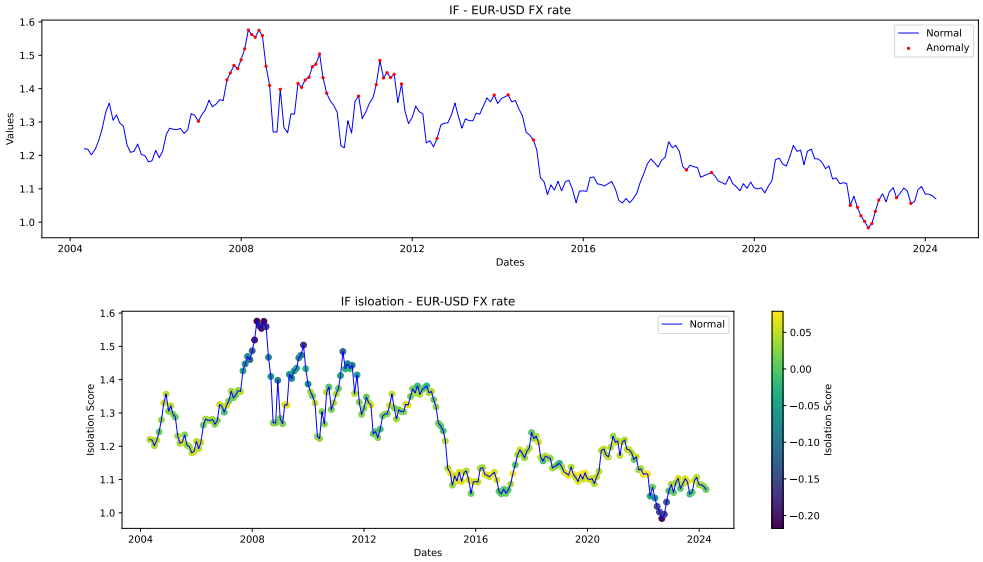

Figure 4.5.4 EUR/USD FX rate anomalies and separation score for Isolation Forest.

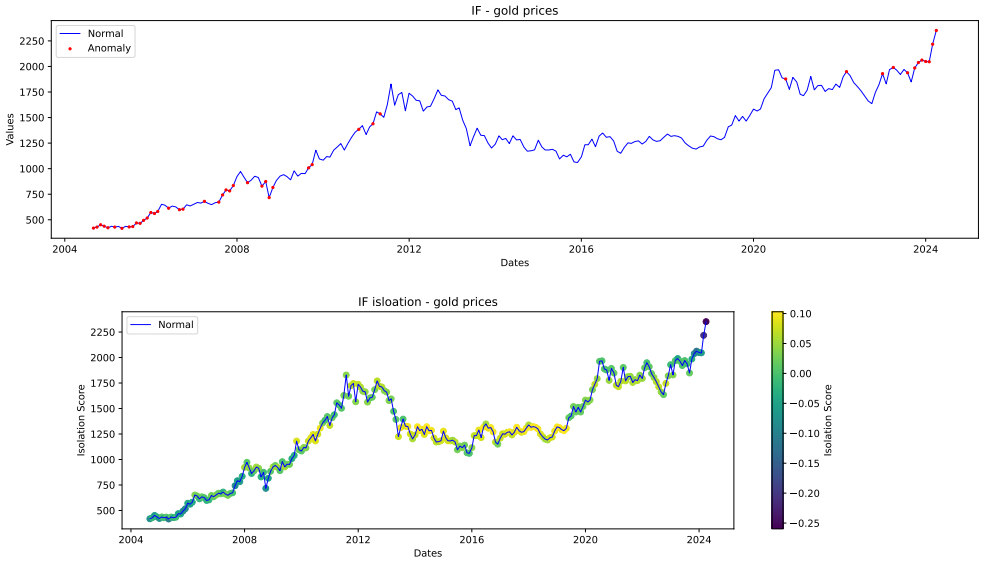

Figure 4.5.5 Gold price anomalies and separation score for Isolation Forest.

Figure 4.5.6 Oil price anomalies and separation score for Isolation Forest.

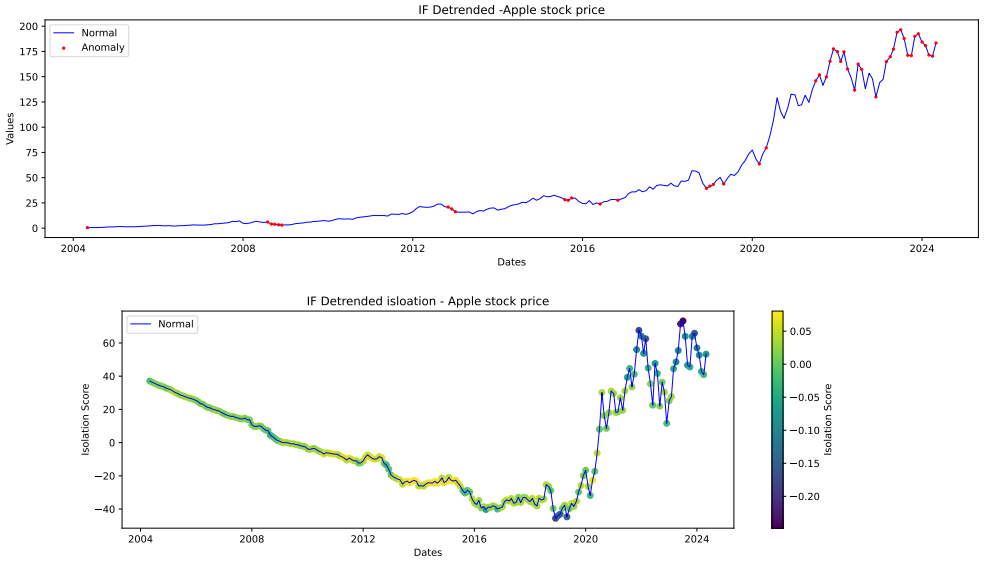

Figure 4.5.7 Apple stock price anomalies and separation score for Isolation Forest with detrended data.

and has allowed the algorithm to showcase its power in anomaly detection better.

In the case of FX rates, the detrending brings us an interesting observation - for the EUR/JPY curve (Figure 4.5.9), the anomaly detection is practically identical to the unadjusted data (see Figure 4.5.3). When compared to the EUR/USD graph (Figure 4.5.10) where the anomaly detection result is quite different from the one for the original data (see Figure 4.5.4), we can deduce, that the trend for the EUR/JPY curve is negligible for the Isolation Forest algorithm.

Interestingly enough, the oil prices anomaly detection in Figure 4.5.12 ended up similarly to the ones for EUR/JPY. When compared to the detection on normal curve in Figure 4.5.6 we can see that the isolation graphs are almost identical and so are the anomaly detections. For the gold in Figure 4.5.11, the detection is still not good, as all of the anomaly points are accumulated around the year 2012, completely ignoring any other market shifts.

4.5.3 Differenced Data

After differencing the data, the Apple stocks anomaly detection in Figure 4.5.13 looks very similar to the one of the differenced Autoencoder from Figure 4.2.13. The isolation graph shows a possible explanation - the initial samples have clear outliers, which are unfortunately overshadowed by the samples from

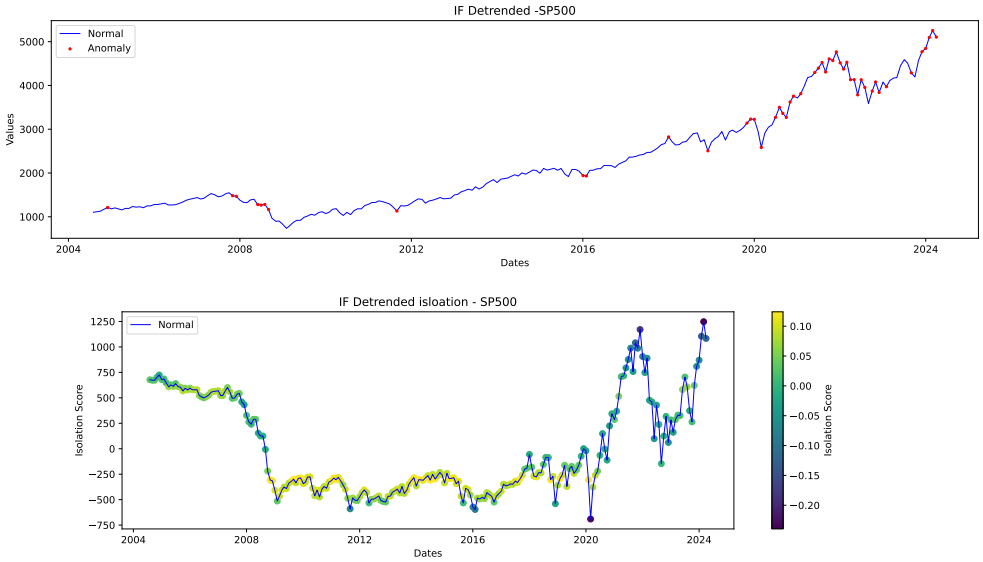

Figure 4.5.8 S&P500 index price anomalies and separation score for Isolation Forest with detrended data.

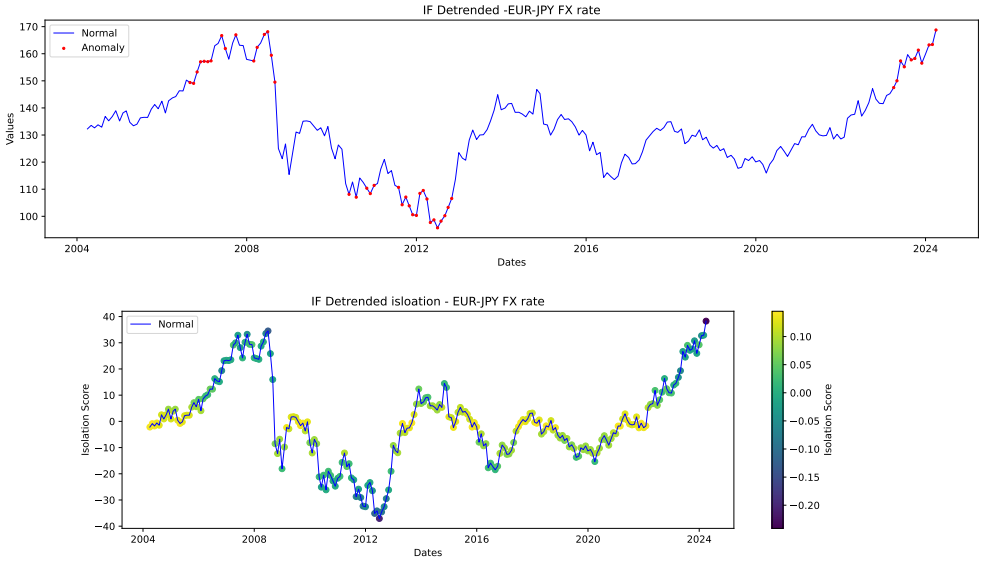

Figure 4.5.9 EUR/JPY FX rate anomalies and separation score for Isolation Forest with detrended data.

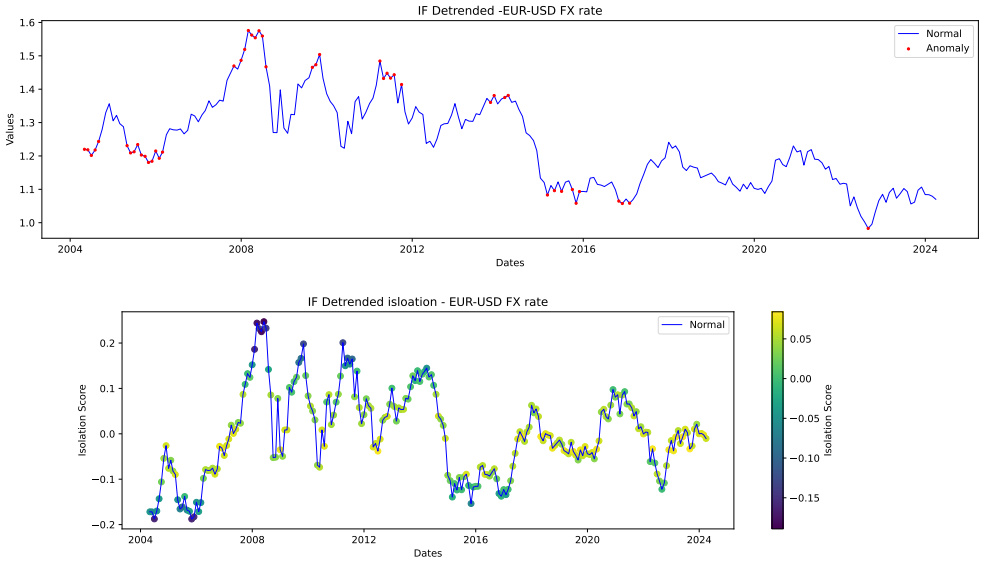

Figure 4.5.10 EUR/USD FX rate anomalies and separation score for Isolation Forest with detrended data.

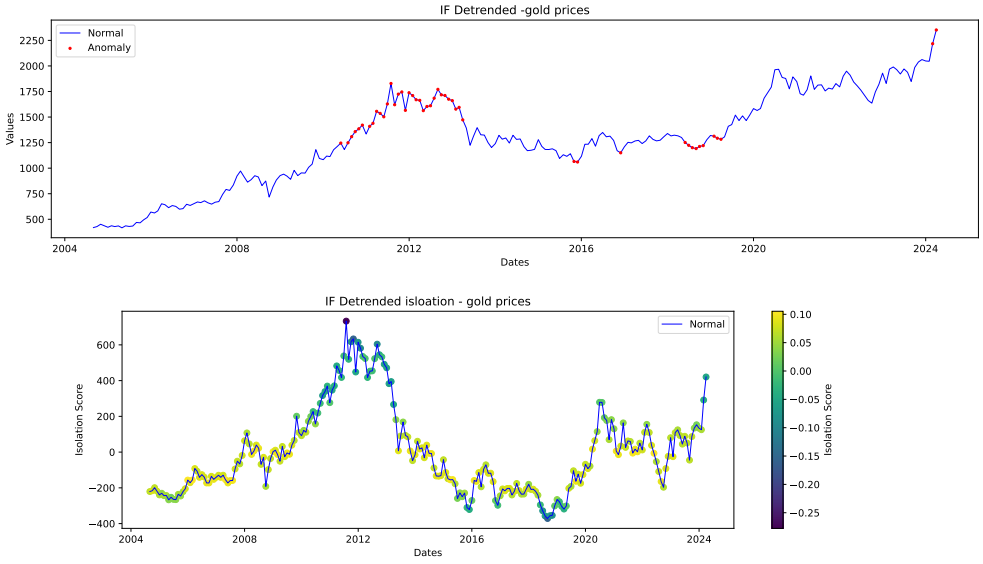

Figure 4.5.11 Gold price anomalies and separation score for Isolation Forest with detrended data.

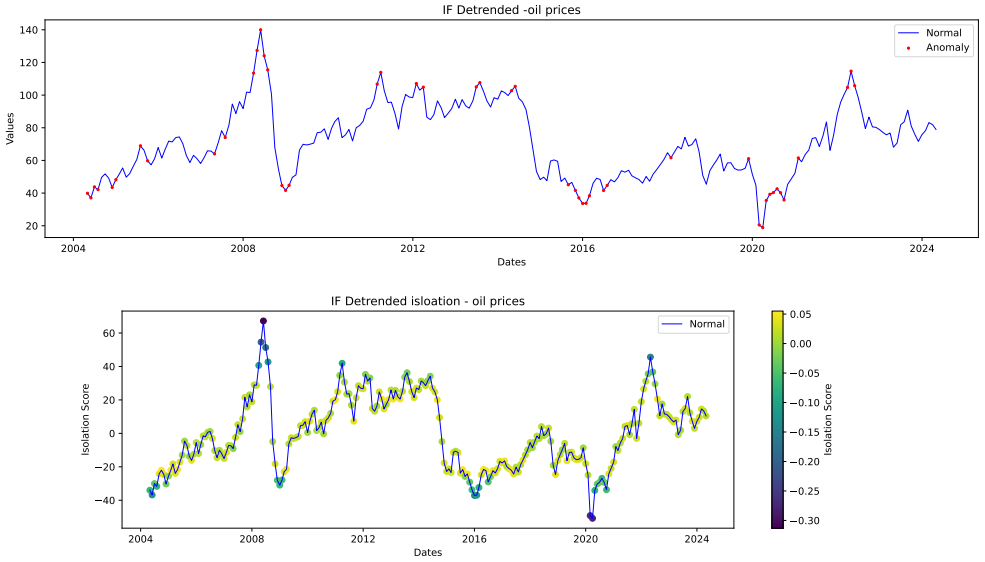

Figure 4.5.12 Oil price anomalies and separation score for Isolation Forest with detrended data.

after 2020, where the absolute difference is several times higher. As for the S&P500 (Figure 4.5.14), we can see that its behavior is similar to the Apple stocks in this case. This suggests that the relative change would be a much better adjustment for this dataset.

The EUR/JPY FX rate, as mentioned earlier, has little to no trend that the Isolation Forest has to deal with. Therefore, for the differenced series, it performs well, as shown in Figure 4.5.15, where we can see that the algorithm determines most significant market shifts, although leaving out the 2020 market crash (even though here we can suspect that the crash did not have a signifficant impact on this rate, since it has not been registered by any of the algorithms, except for Hurst Exponent in Figure 4.4.3). As for the EUR/USD FX rate, the algorithm works very well, with almost all of the major market shifts highlighted, as shown in 4.5.16.

Commodities, when differenced, show again very similar results for the Isolation Forests as for the Autoencoder as observed in Figures 4.5.18 and 4.5.17 when compared to Figures 4.2.18 and 4.2.17.

4.5.4 Data with relative change

The Apple stock anomaly detection in this case (Figure 4.5.19) also resembles the results of the Autoencoder on a similar dataset (see Figure 4.2.19), however unlike Autoencoder, it also picks up the anomalies during the year 2022, where

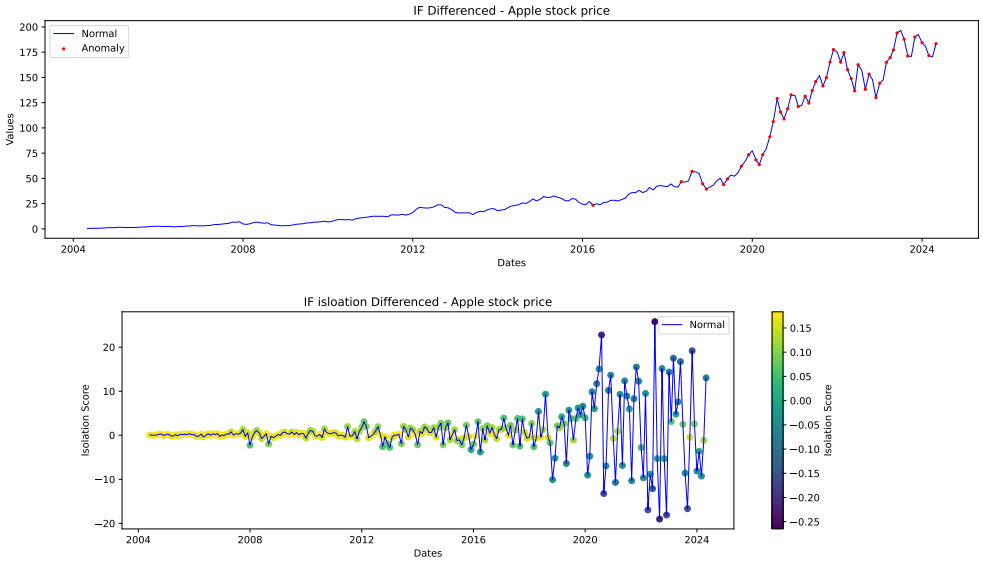

Figure 4.5.13 Apple stock price anomalies and separation score for Isolation Forest with differenced data.

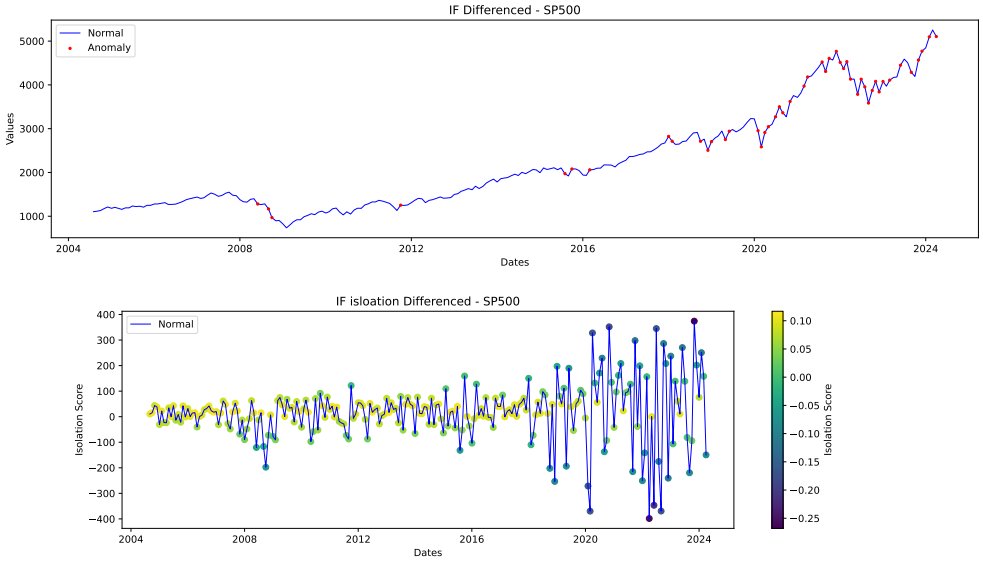

Figure 4.5.14 S&P500 index price anomalies and separation score for Isolation Forest with differenced data.

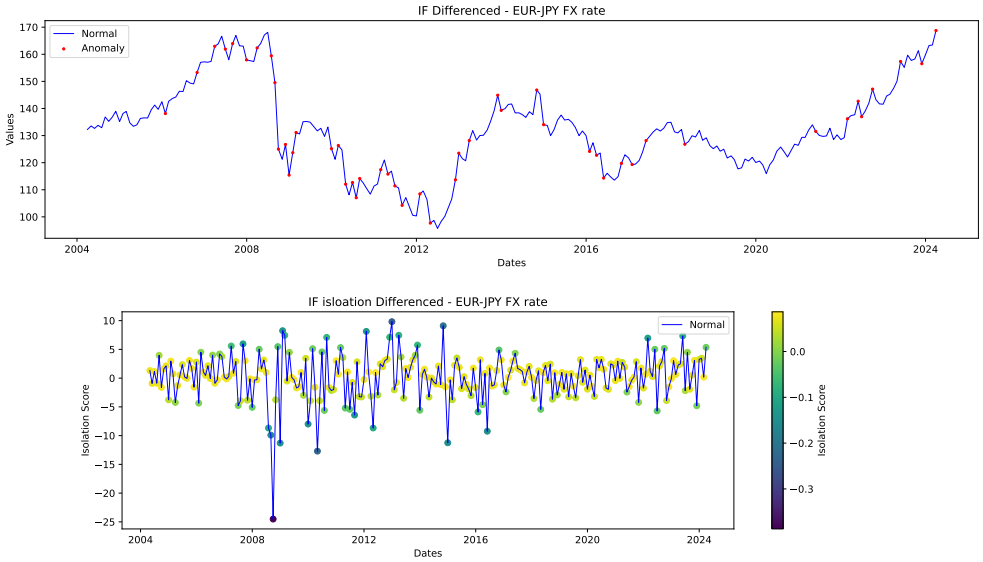

Figure 4.5.15 EUR/JPY FX rate anomalies and separation score for Isolation Forest with differenced data.

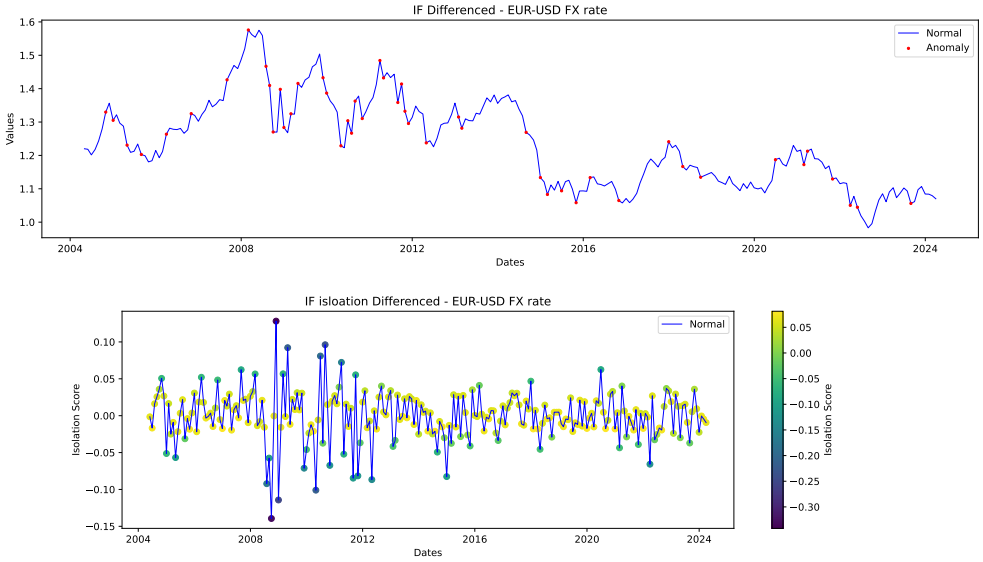

Figure 4.5.16 EUR/USD FX rate anomalies and separation score for Isolation Forest with differenced data.

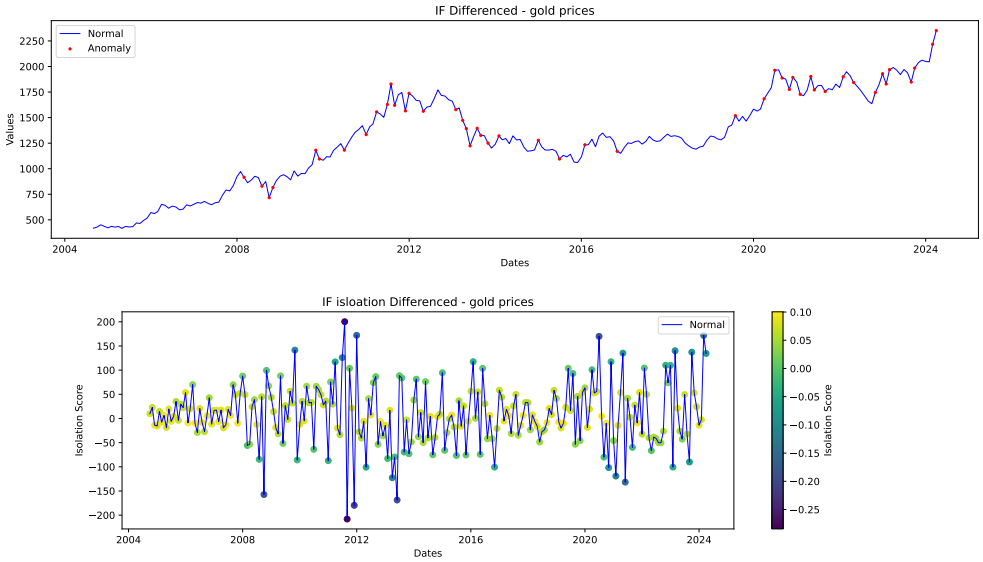

Figure 4.5.17 Gold price anomalies and separation score for Isolation Forest with differenced data.

Figure 4.5.18 Oil price anomalies and separation score for Isolation Forest with differenced data.

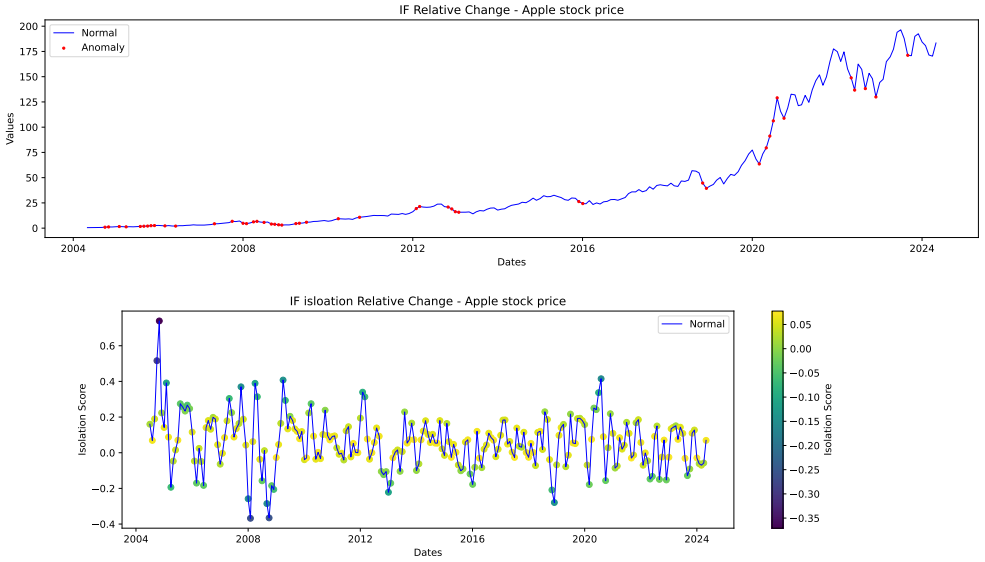

Figure 4.5.19 Apple stock price anomalies and separation score for Isolation Forest with relative change time series.

there was a recession due to the Covid-19 pandemics related inflation. In this case, the Isolation Forest performs remarkably well and highlights the most significant market capitalization shifts related to Apple stocks for the last two decades. Also, for the S&P500, the Isolation Forests give us solid results as seen in Figure 4.5.20, pointing out most of the relevant shifts, with the only bigger one left out was 2018, also called the ”worst year for S&P500 after the 2009 crisis”. However, looking at the isolation score graph, we can see that the shift is marked as significant, and if we lowered a threshold, it would be one of the first samples to be marked as an anomaly.

As for the FX rates, looking at Figures 4.5.15 and 4.5.16, they are again very similar to Figures 4.5.15 and 4.5.16. For the EUR/JPY curve, the increase after 2012 is highlighted more, but some bigger peaks are not labeled as anomalies when we compare it to the Isolation Forest on a differenced series. For the EUR/USD, the dip in 2016 and peak in 2018 are missing when compared to the differenced series again.

For commodities, we can see from Figures 4.5.24 and 4.5.23 that the isolation score graphs are composed differently, with the relative changes focusing less on the year 2012 for the gold prices and the emphasis is put on the 2020 market crash for the oil prices. Overall, the detections are slightly worse for the commodities when we compare the performance to the differenced series.

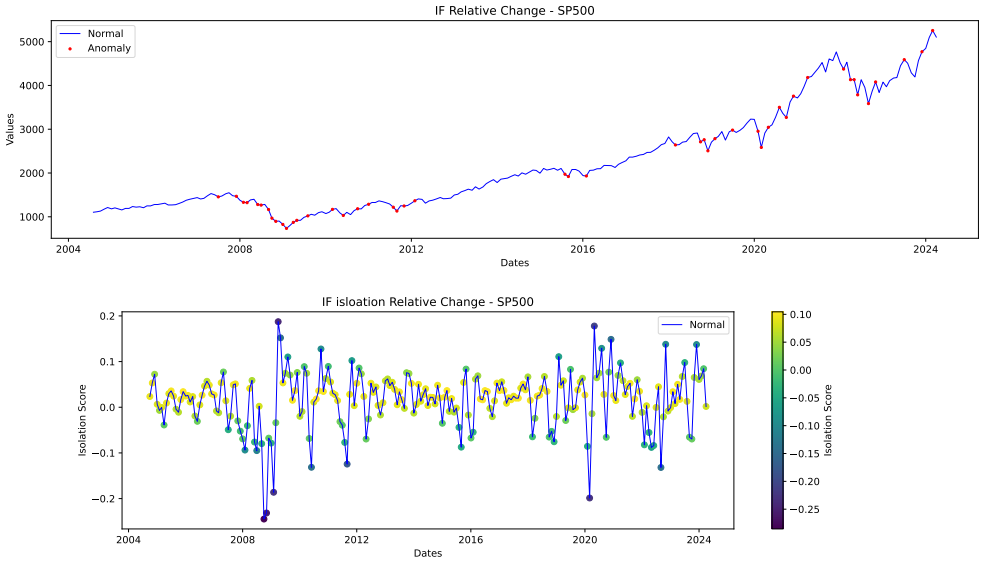

Figure 4.5.20 S&P500 index price anomalies and separation score for Isolation Forest with relative change time series.

Figure 4.5.21 EUR/JPY FX rate anomalies and separation score for Isolation Forest with relative change time series.

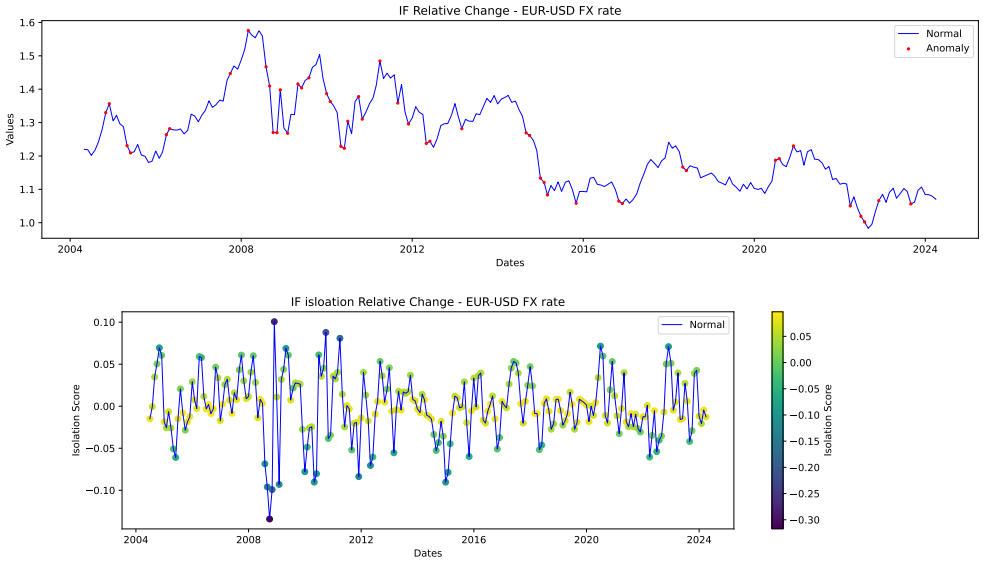

Figure 4.5.22 EUR/USD FX rate anomalies and separation score for Isolation Forest with relative change time series.

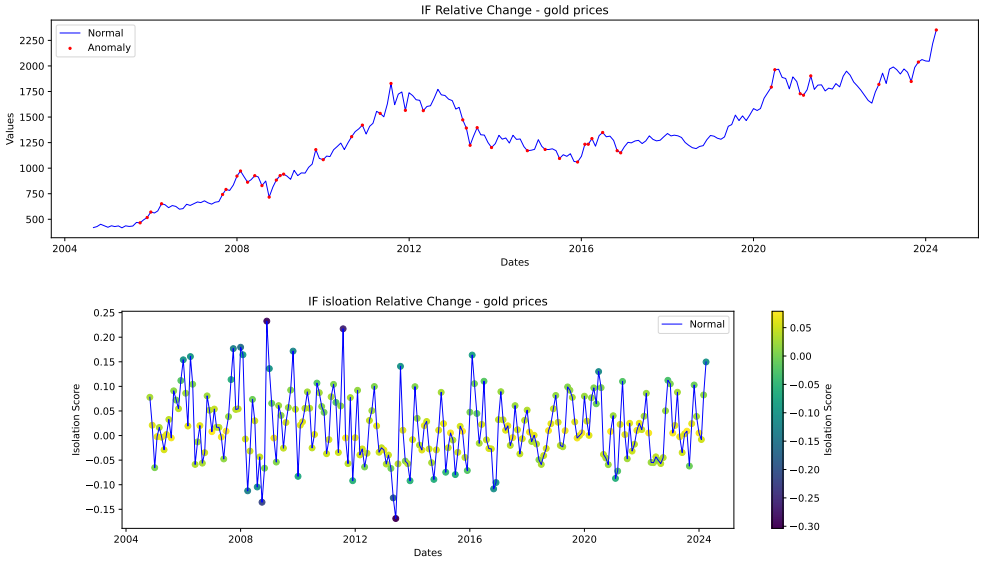

Figure 4.5.23 Gold price anomalies and separation score for Isolation Forest with relative change time series.

Figure 4.5.24 Oil price anomalies and separation score for Isolation Forest with relative change time series.

4.5.5 Conclusion

The Isolation Forests performed remarkably well, especially on the differenced data. The algorithms did highlight the most important shifts and are pretty suitable for all of the product classes. Although the detections were not great with normal data, the differencing makes a great difference for the Isolation Forest algorithm.

4.6 SVM anomaly detection

4.6.1 Normal Data

On the stocks, the SVM shows similar behavior as the Isolation Forest or Autoencoder, as we can see in Figures 4.6.1 and 4.6.2. The only anomalies being detected in the lowest and most upper parts of the graph. As for the FX rates (Figures 4.6.3 and 4.6.4), the detection again is comparable with the Autoencoder. We can see that the seasonality in these time series and the trends in the equity time series both have significant influences on the detection. The gold prices (Figure 4.6.5) suffer from the exact same behavior as equities or FX rates. In the oil prices graph (Figure 4.6.6), the issue is still visible; however, due to the absence of trends, the algorithm is able to capture more anomalies than in previous classes.

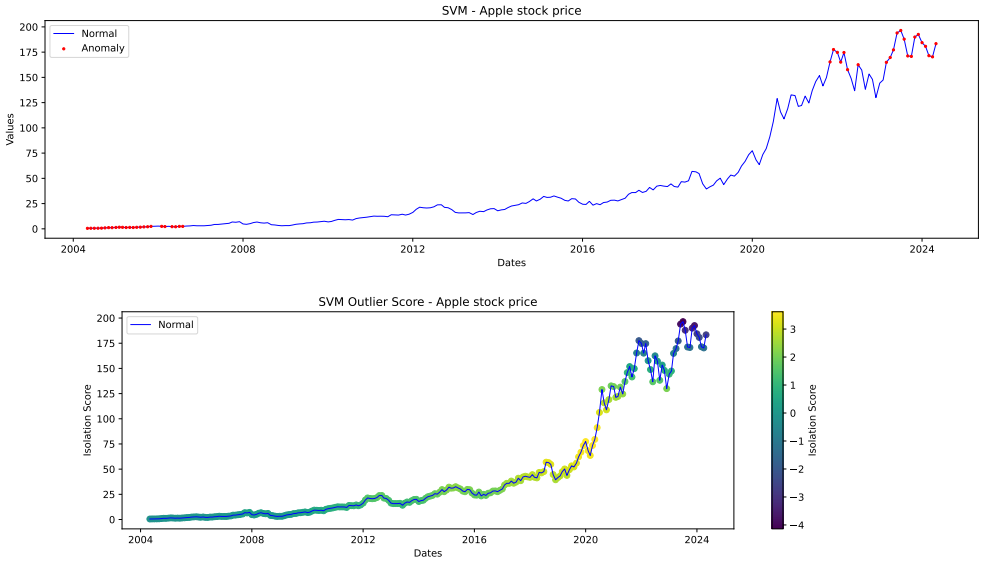

Figure 4.6.1 Apple stock price anomalies and Isolation Score for SVM.

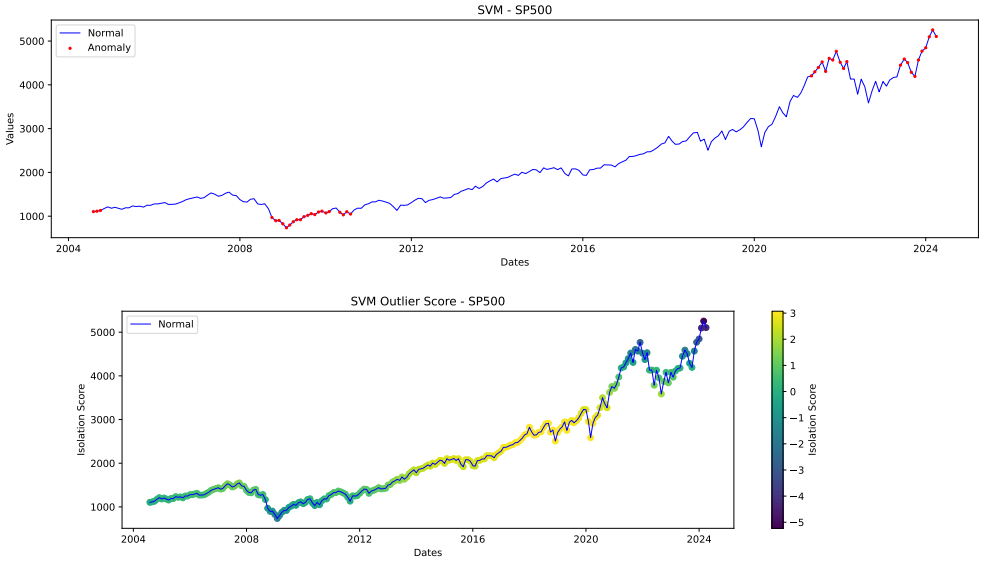

Figure 4.6.2 S&P500 index price anomalies and Isolation Score for SVM.

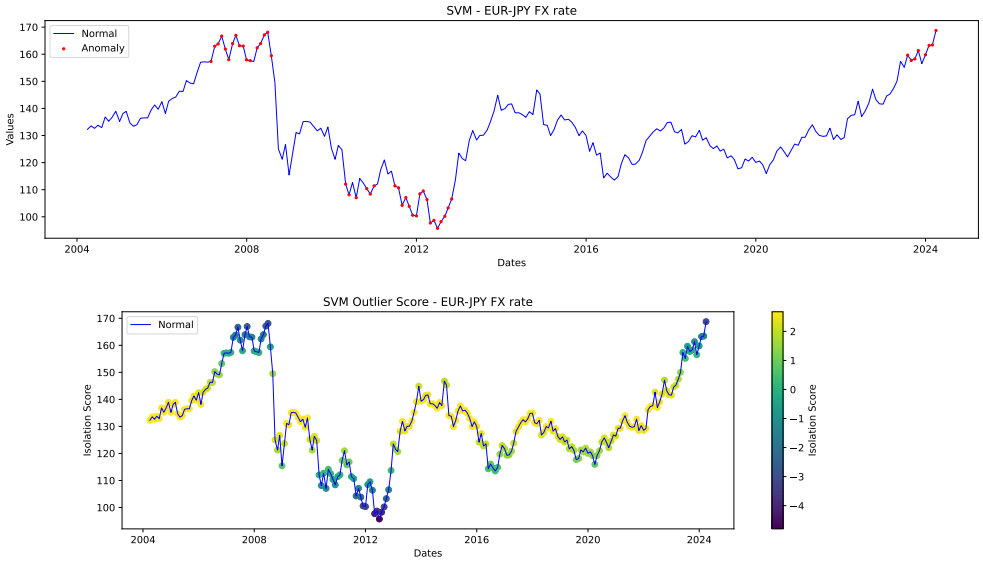

Figure 4.6.3 EUR/JPY FX rate anomalies and Isolation Score for SVM.

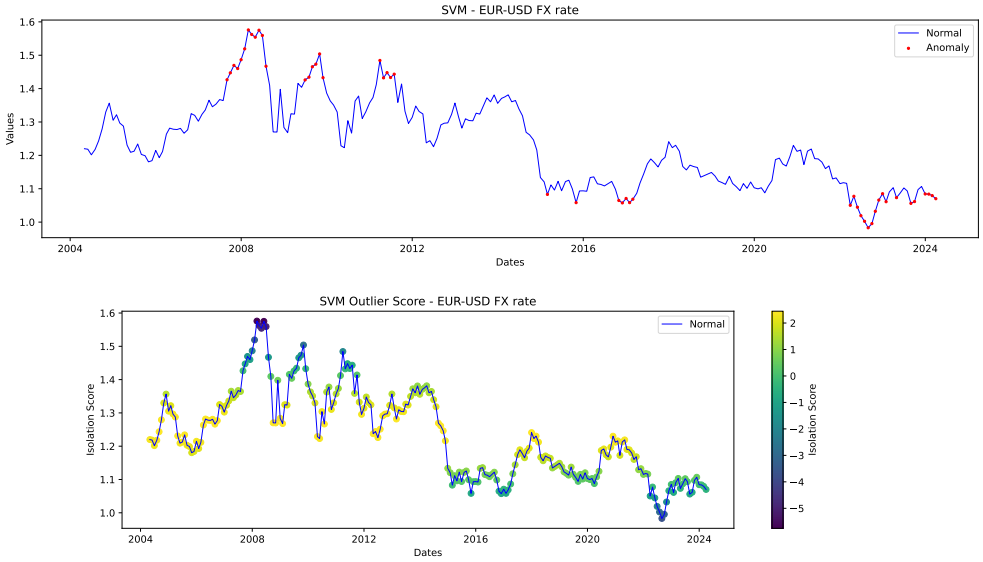

Figure 4.6.4 EUR/USD FX rate anomalies and Isolation Score for SVM.

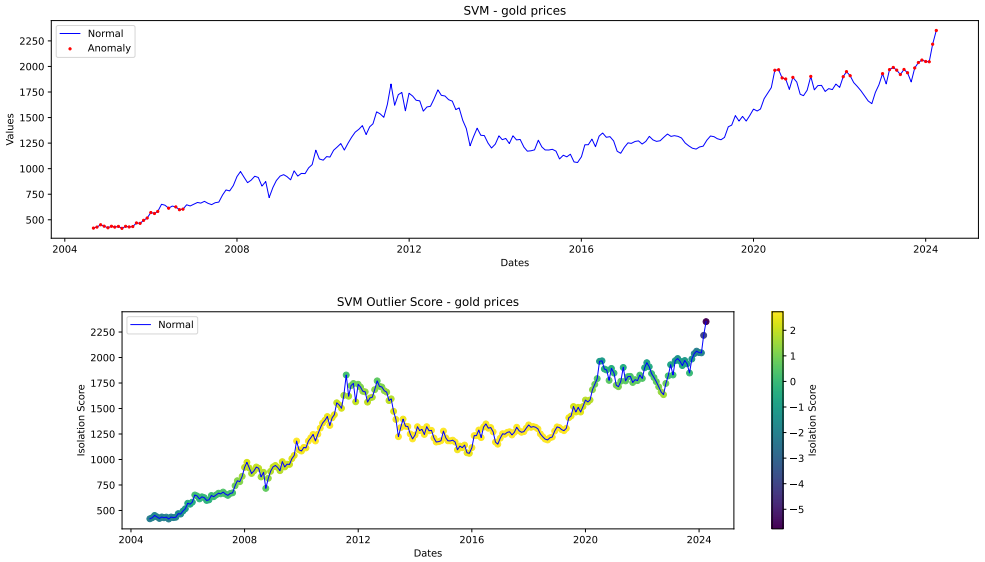

Figure 4.6.5 Gold price anomalies and Isolation Score for SVM.

Figure 4.6.6 Oil price anomalies and Isolation Score for SVM.

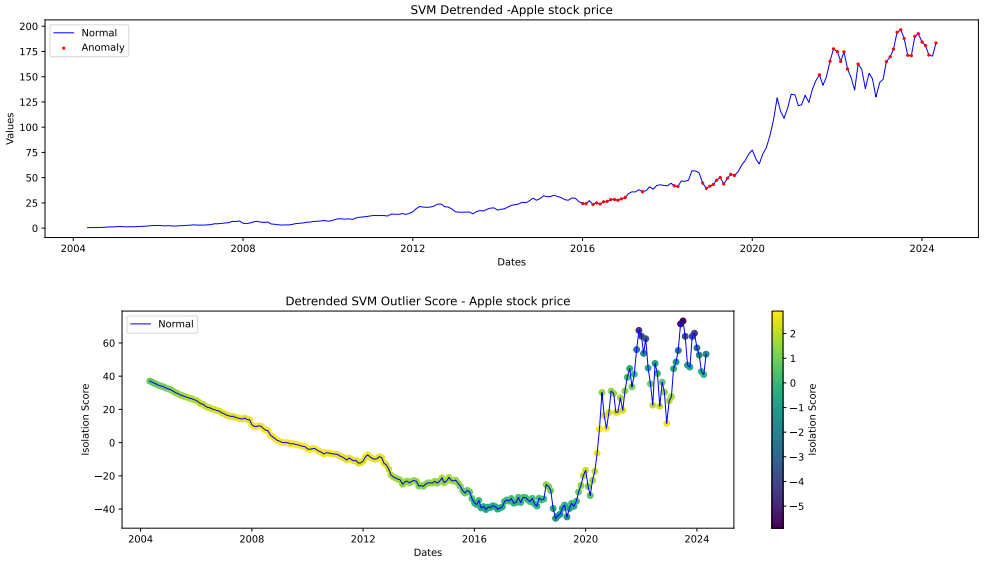

Figure 4.6.7 Apple stock price anomalies and Isolation Score for SVM with detrended data.

4.6.2 Detrended Data

Getting rid of the trend did not particularly help the detection of the Apple stocks (Figure 4.6.7); it just removed the anomaly labels from the beginning and moved them toward the years after 2020. On the S&P500, however, the detection has improved, with the points mainly concentrating on significant market shifts.

As for the FX rates, the EUR/JPY curve did not change much, since the trend is not present here, as we can see in Figure 4.6.9. Neither did the EUR/USD FX rates in Figure 4.6.4.

For the oil prices, again, the algorithm did not react differently; the detections in Figures 4.6.12 and 4.6.6 are basically the same. Where the detrending had some effect are the gold prices in Figure 4.6.11. Here, we can see that contrary to the original data in Figure 4.6.5, the detections are now mostly announced in the middle part of the graph. The detection still is not perfect, but it has shown some improvement.

4.6.3 Differenced Data

For the differenced series of Apple stocks (Figure 4.6.13) the result looks exactly the same as it does for Autoencoder (Figure 4.2.13), LSTM (Figure 4.3.13) and Isolation Forest (Figure 4.5.13). This may be because differencing the Apple stocks brings the focus on the last years, where the absolute differ-

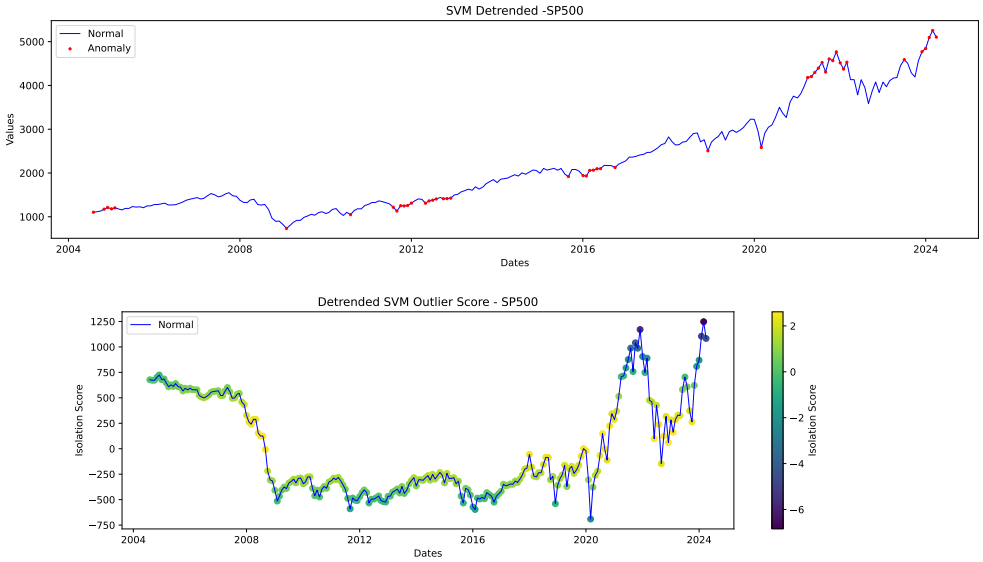

Figure 4.6.8 S&P500 index price anomalies and Isolation Score for SVM with detrended data.

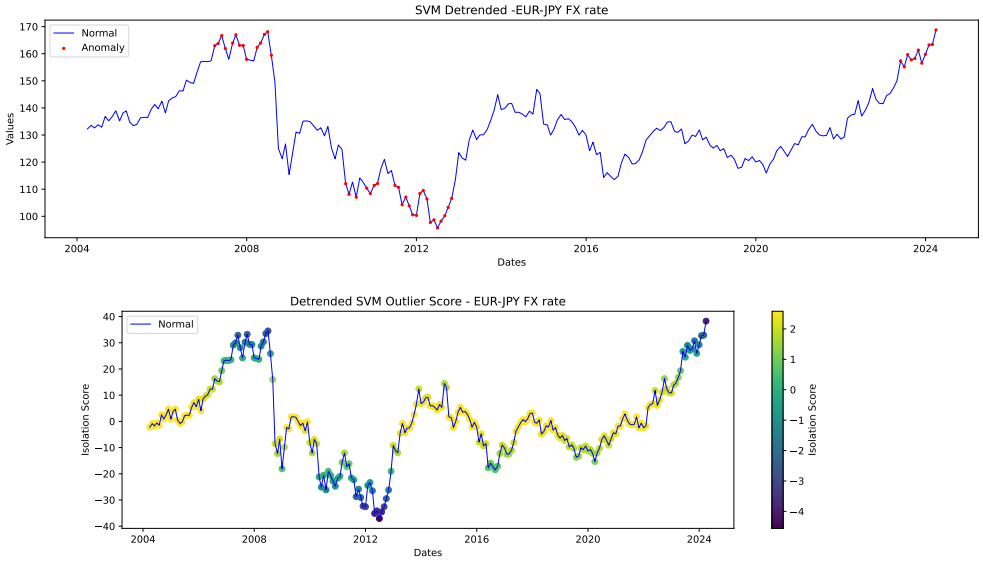

Figure 4.6.9 EUR/JPY FX rate anomalies and Isolation Score for SVM with detrended data.

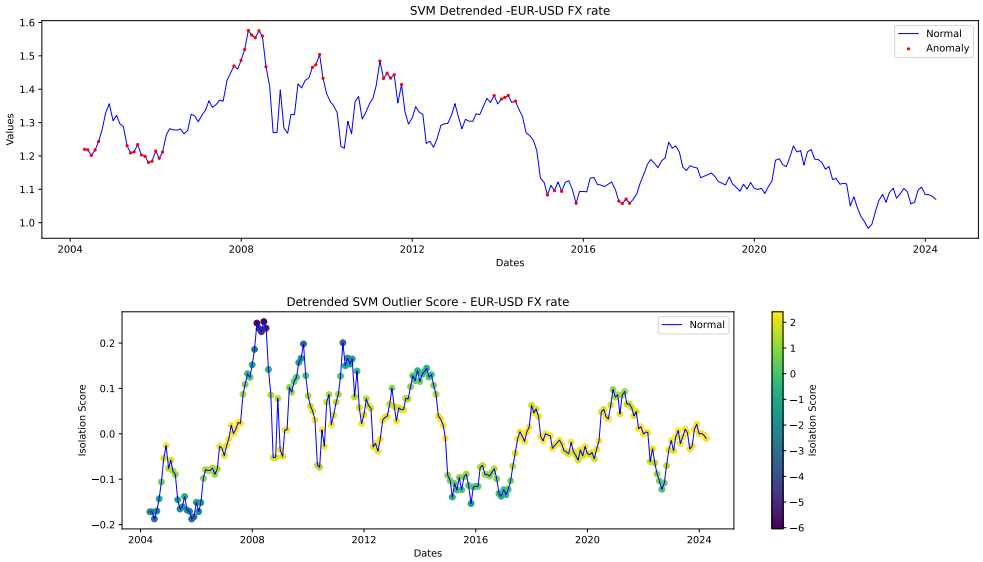

Figure 4.6.10 EUR/USD FX rate anomalies and Isolation Score for SVM with detrended data.

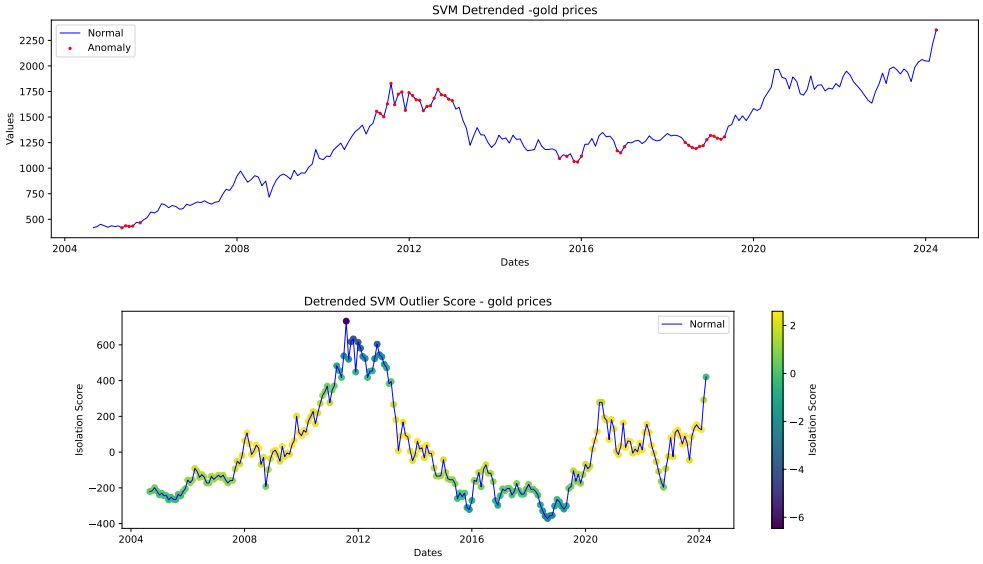

Figure 4.6.11 Gold price anomalies and Isolation Score for SVM with detrended data.

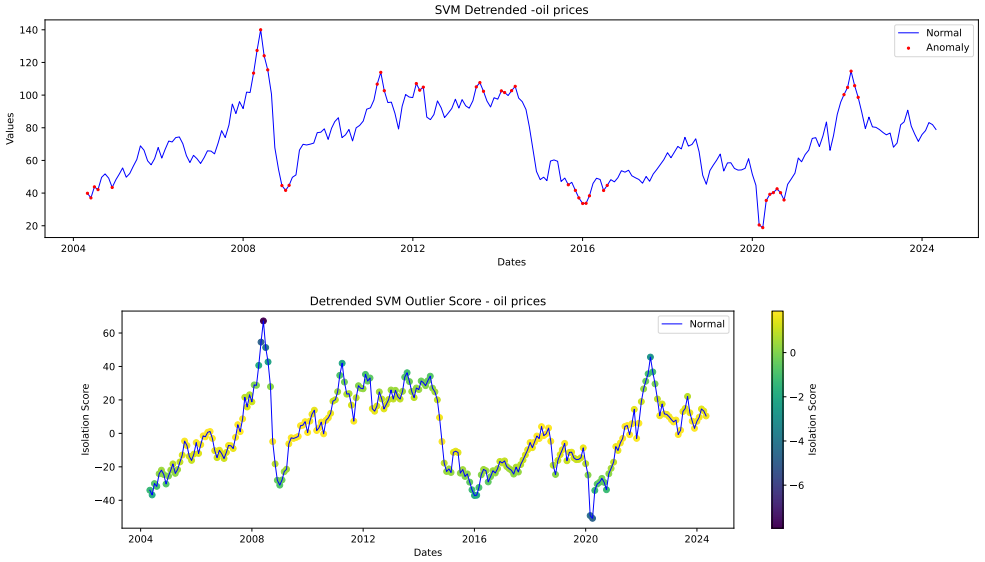

Figure 4.6.12 Oil price anomalies and Isolation Score for SVM with detrended data.

ences are much higher than years ago due to the enormous growth of Apple Inc. As for the S&P500 (Figure 4.6.14), the situation is similar; the growth of S&P500 was not as striking as in the case of Apple, but still, the capitalization is almost 500% of where it was in the year 2004. For the EUR/JPY FX rate, the results are almost the same as the ones from the Isolation Forests (see Figures 4.6.15 and 4.5.15). The same case is the EUR/USD curve (see Figures 4.6.16 and 4.5.16). However, that is not necessarily wrong since, in both cases, the detection is pretty solid and highlights almost all significant shifts. The commodities are also similar in this regard (see Figures 4.6.17 and 4.6.18 when compared to Figures 4.5.17 and 4.5.18). For the different series, the Isolation Forests and SVM work very similarly and give comparable detections.

4.6.4 Data with relative change

As for the relative change time series, the prediction on the Apple stocks is very good, as we can see in Figure 4.6.19. The prediction closely reminds the Isolation Forest one (see Figure 4.5.19), however here we can see more events covered. The same case is the S&P500, where the SVM performs in similar way to the Isolation Forest (see Figures 4.6.20 and 4.5.20).

For both FX rates, the JPY/EUR curve in Figure 4.6.21 has similar anomalies to the one we have seen in the case of differenced series (see Figure 4.6.15), however, we can see that the algorithm now picks up more anomalous points,

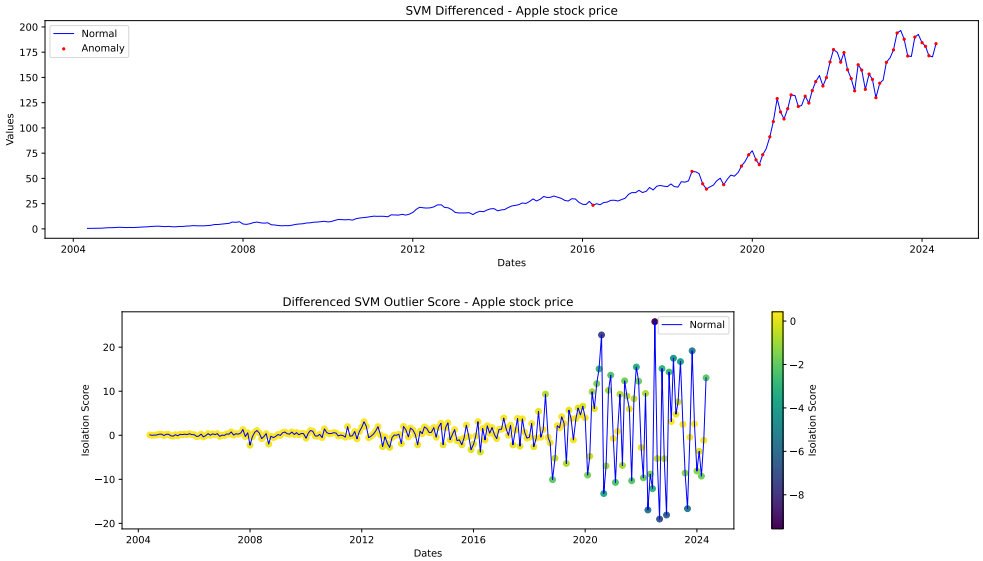

Figure 4.6.13 Apple stock price anomalies and Isolation Score for SVM with differenced data.

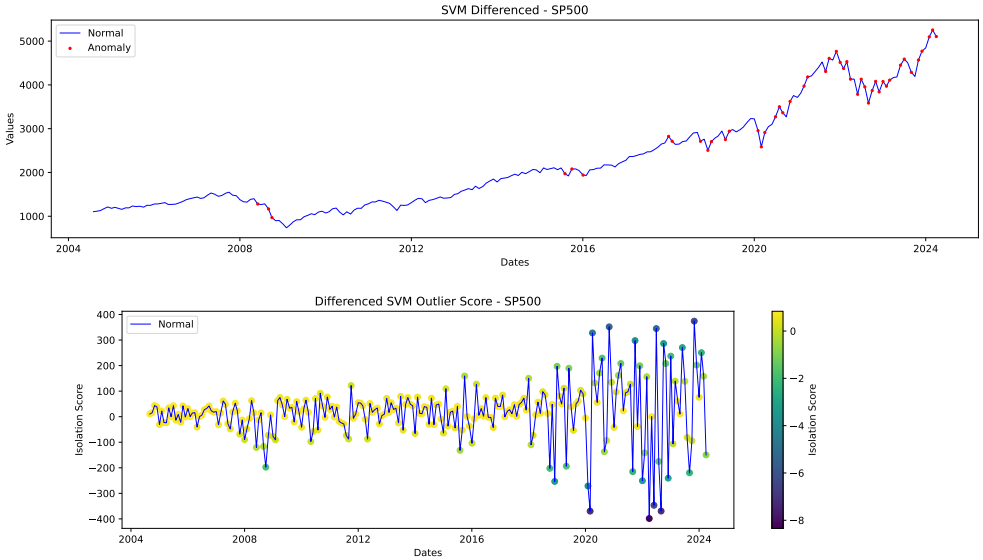

Figure 4.6.14 S&P500 index price anomalies and Isolation Score for SVM with differenced data.

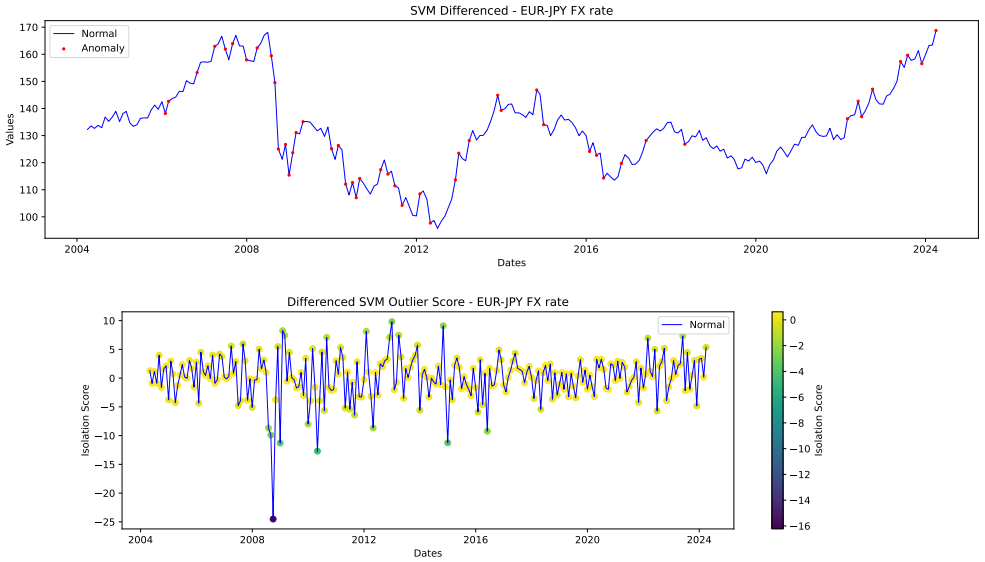

Figure 4.6.15 EUR/JPY FX rate anomalies and Isolation Score for SVM with differenced data.

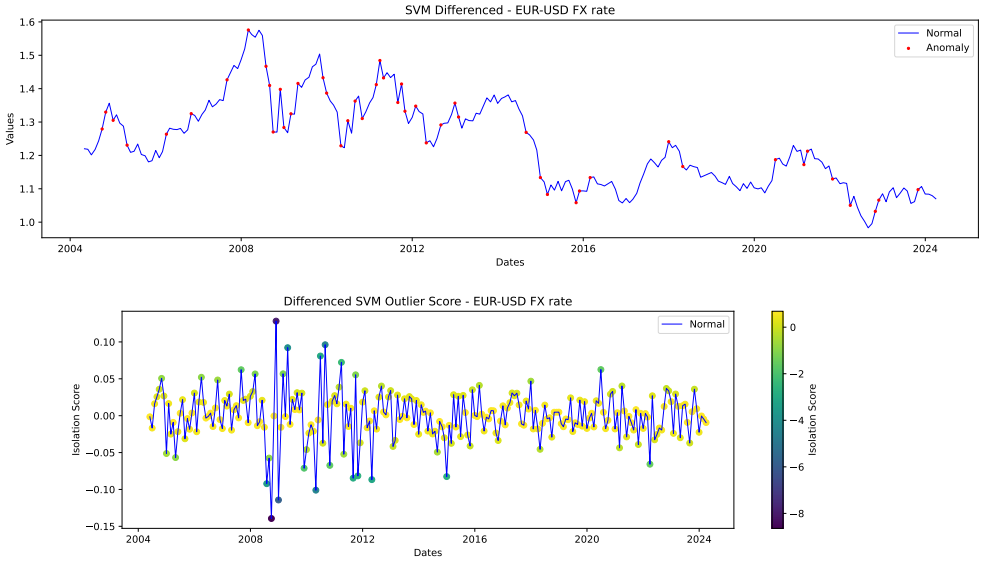

Figure 4.6.16 EUR/USD FX rate anomalies and Isolation Score for SVM with differenced data.

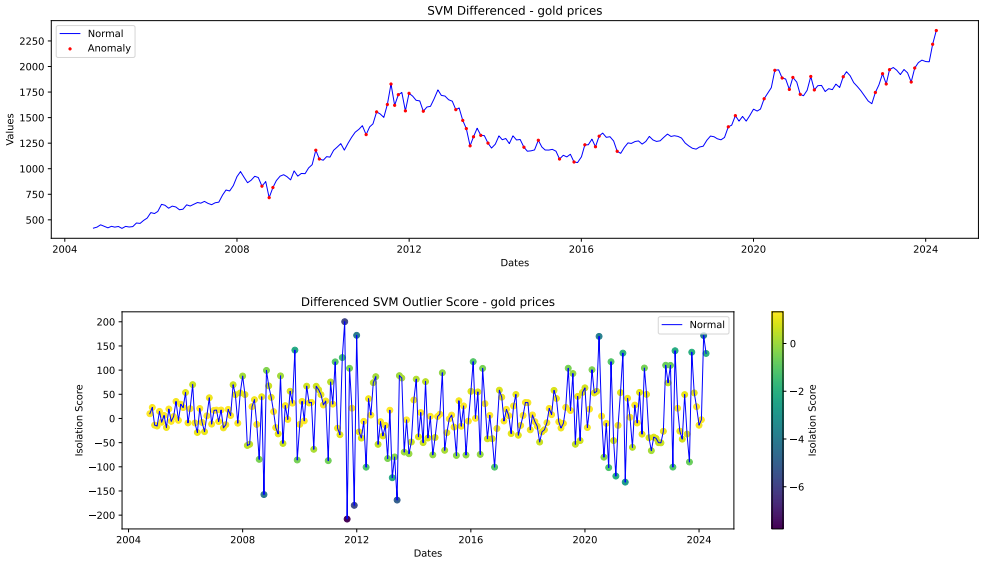

Figure 4.6.17 Gold price anomalies and Isolation Score for SVM with differenced data.

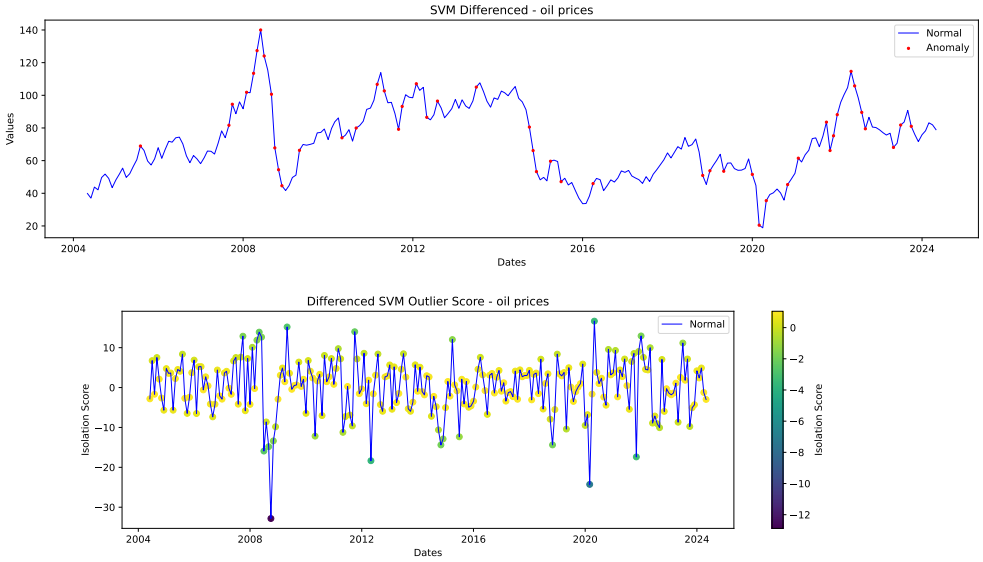

Figure 4.6.18 Oil price anomalies and Isolation Score for SVM with differenced data.

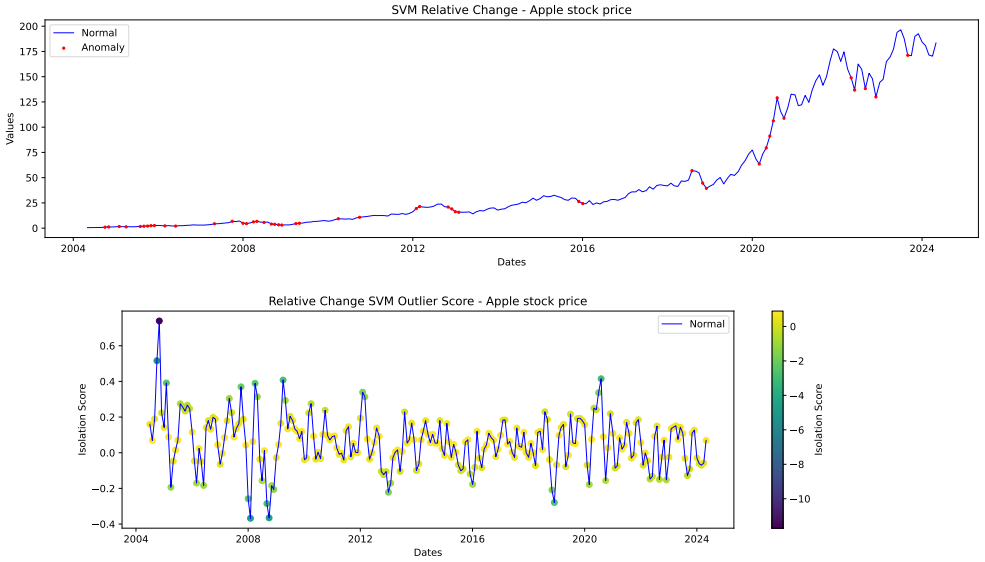

Figure 4.6.19 Apple stock price anomalies and Isolation Score for SVM with relative change time series.

for example there is new detection on the peak between the years 2016 and 2017. The USD/EUR anomaly detection in Figure 4.6.22 looks also similar to the diffrenced series (see Figure 4.6.16).

The detection on gold prices in Figure 4.6.23 shows certain improvement against the differenced data (see Figure 4.6.17). We can see that now newly we have highlighted the peaks that were preceeding the market crash in 2008, as well as detections in the period before 2008, with less detections after the 2020. For oil in Figure 4.6.24, the situation is parallel; if we look at Figure 4.6.18, we can see, that for relative changes the detections are more concentrated around 2016, with dominant shift being the 2020.

4.6.5 Conclusion

In conclusion we can see that the SVM performs very well, and its detections are resembling those of Autoencoder and Isolation Forest in many cases. It handles very well the stock anomaly detection and does not do bad in the other cases. The best way to use this algorithm is with the relative change data, as the isolation of the outlier becomes more apparent to the alogrithm.

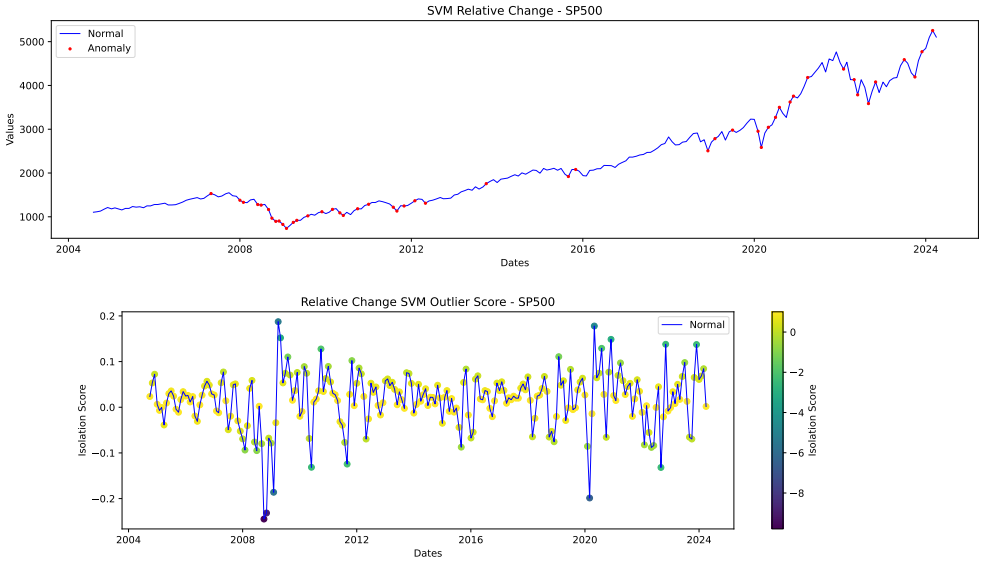

Figure 4.6.20 S&P500 index price anomalies and Isolation Score for SVM with relative change time series.

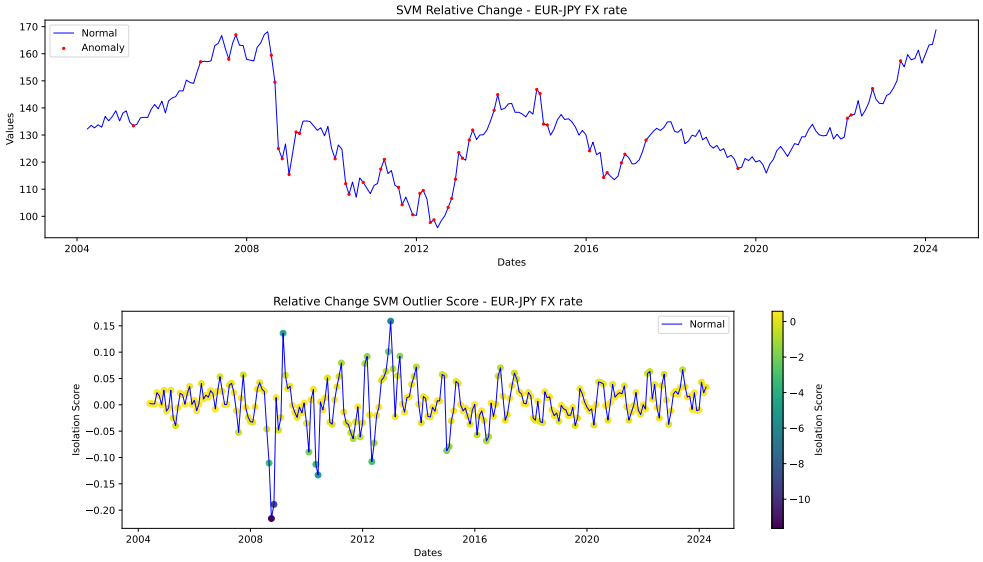

Figure 4.6.21 EUR/JPY FX rate anomalies and Isolation Score for SVM with relative change time series.

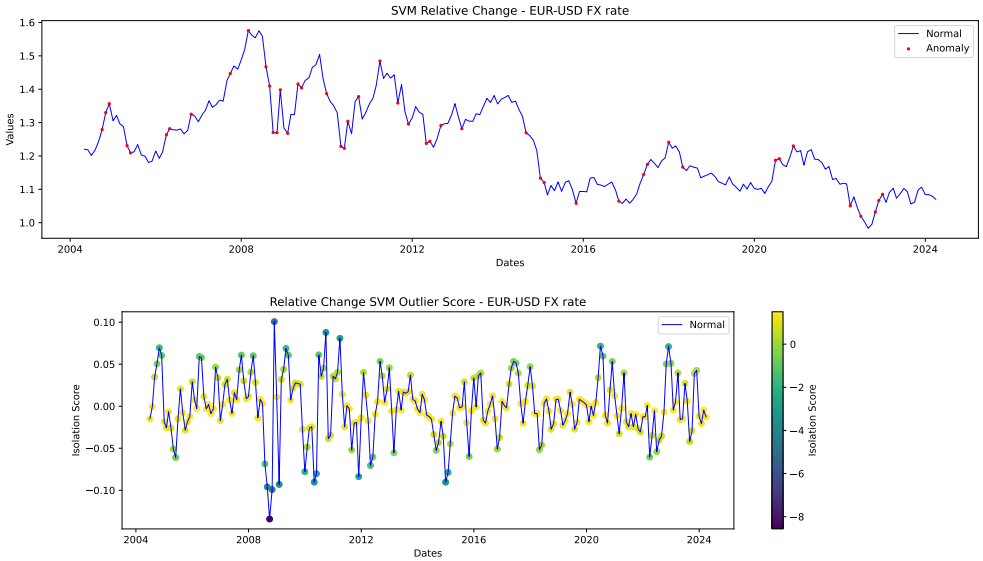

Figure 4.6.22 EUR/USD FX rate anomalies and Isolation Score for SVM with relative change time series.

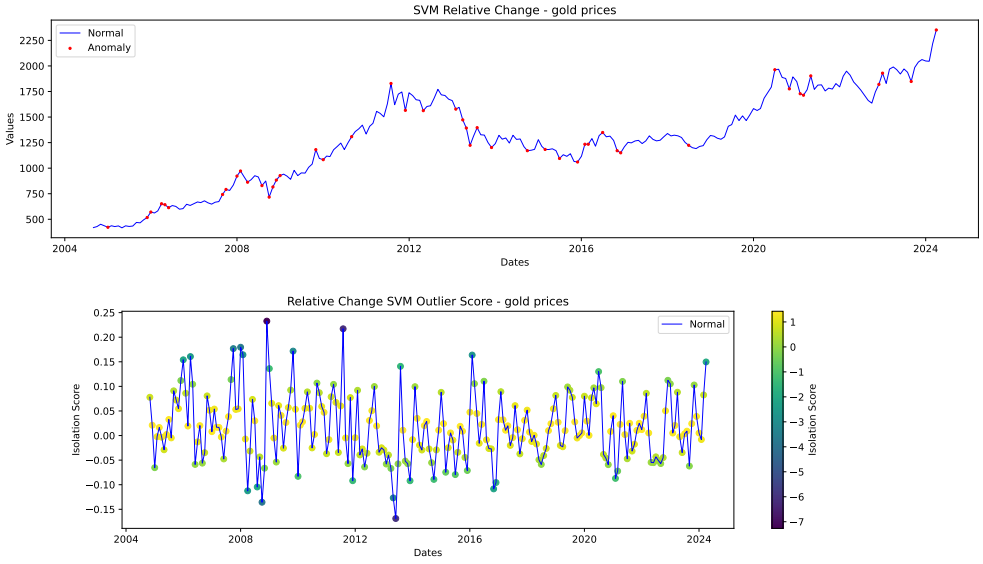

Figure 4.6.23 Gold price anomalies and Isolation Score for SVM with relative change time series.

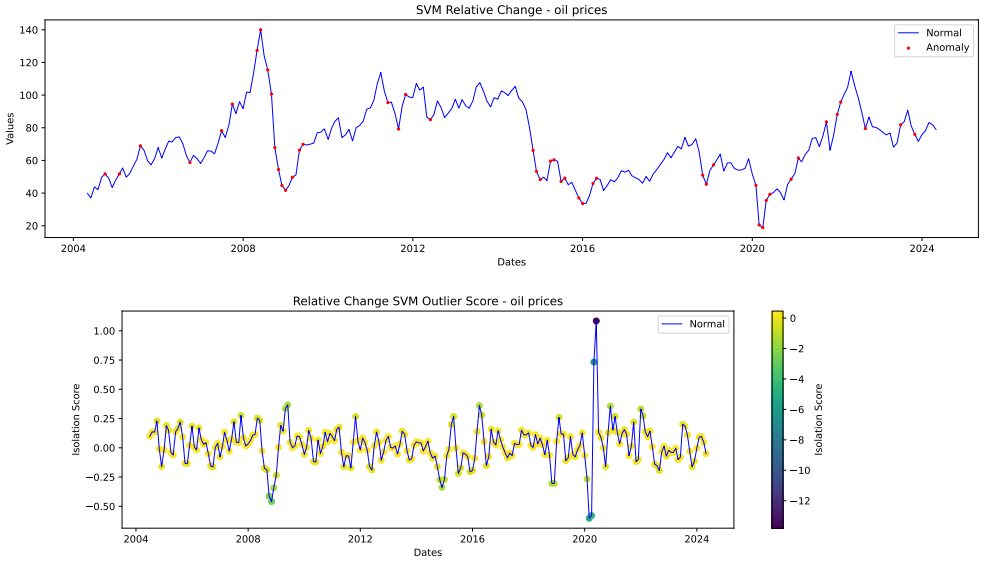

Figure 4.6.24 Oil price anomalies and Isolation Score for SVM with relative change time series.

4.7 Retraining algorithms

In [1], the authors used sequential retraining with several strategies. They worked with the already mentioned interest rate yield curves, which had five tenors and 50 pillars. However, these data were not available to the author of this thesis. Instead, the retraining was tried on S&P500, on normal and differenced data. The retraining method was sliding windows with various sizes; in this thesis, we will use a graph with a window size of 30.

From Figures 4.7.1 and 4.7.2, we can see that the results do not correspond to what the authors of the original research experienced. The reconstruction and fitting are not very good, and this can mainly be attributed to a small number of samples because the authors of the original thesis had more detailed data available. We could see that the reconstruction is not good, even for the differenced data, and the detection is mediocre at best. Therefore, for further research, we will continue with only single-time trained models since this is more computationally feasible and the results are similar.

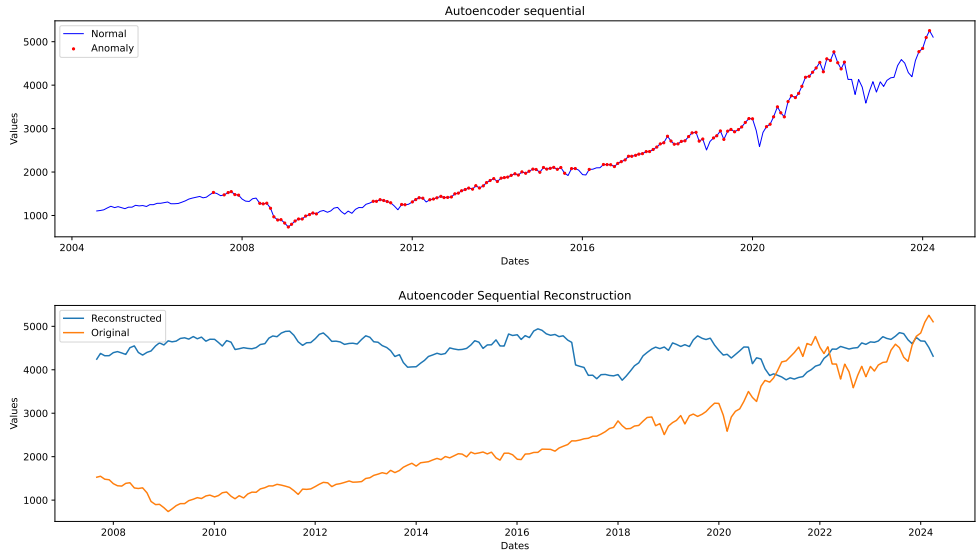

Figure 4.7.1 Retraining on Autoencoder.

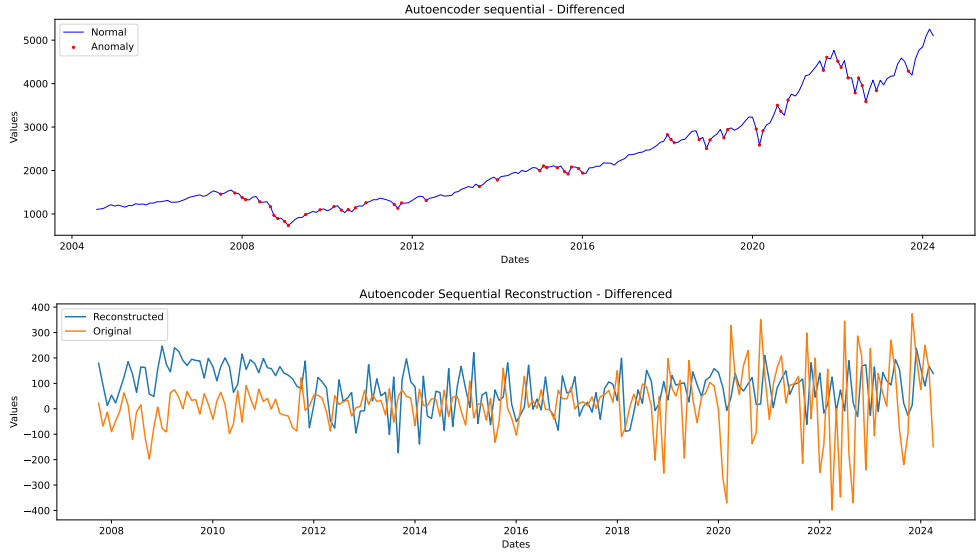

Figure 4.7.2 Retraining on Autoencoder with differenced data.

4.8 Artifficial anomalies

In previous sections, we have seen the algorithms perform on various financial products and described their performance. However, this observation was only done with expert knowledge of the underlying processes and was more focused on determining which algorithm works best on which data. In this section, we will create an artificial time series that contains handmade anomalies. The base time series $S(t)$ is generated as:

$$S(t) = (\text{base\_trend\_slope} \times t) + \sin\left(\frac{2\pi t}{\text{period\_length}}\right) + \epsilon_t \tag{4.4}$$

where t is the time step and $\epsilon_t \sim \mathcal{N}(0, \text{noise\_level})$.

The **point anomalies** are then brought in by modifying the series at random points:

$$S(t_i) = S(t_i) \pm \text{anomaly\_magnitude} \times (1 + \text{rand}()). \tag{4.5}$$

for i in the set of point anomaly indices, where rand() creates a random number between 0 and 1.

The **structural breaks** are then made as:

$$S(t) = S(t) + \Delta \text{ for all } t \geq t_j, \tag{4.6}$$

where t_j is the time index starting from which the structural break applies, and Δ (the magnitude of change) is defined as:

$$\Delta = \text{struct\_break\_magnitude} \times (1 + \text{rand}()) \times (\pm 1). \tag{4.7}$$

Here, (± 1) randomly decides the direction of the change (up or down).

The number of data points was set to 500, the anomaly and structural break magnitudes were set to 3, the base trend slope initiated as 0.05, and the period length was set to 30 to simulate monthly seasonality. The number of both the generated anomalies and the structural breaks was set to 2% each. The sensitivity of the algorithm was set to mark the top 5% anomalous points.

In Figure 4.8.1, we can see the resulting graph with the anomalies highlighted.

When we just ran the algorithm with data as they were (see Figure 4.8.2), we noticed that most of the algorithms detected only one or two anomalies, except for the Autoencoder, where the success rate was higher than 50%.

For the detrended data (see Figure 4.8.3), the algorithms did not generally perform well. This may be because the detrending contributed to the disrup-

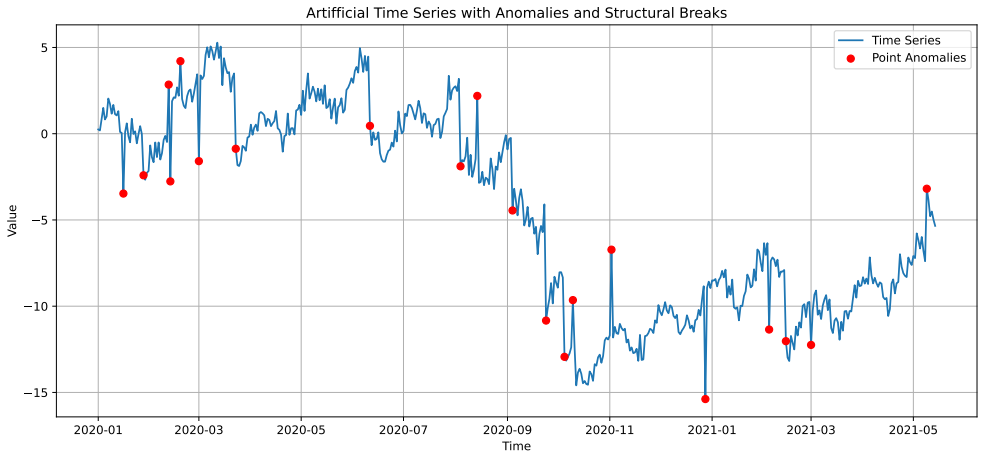

Figure 4.8.1 graph depicting the generated series with highlighted anomaly labels.

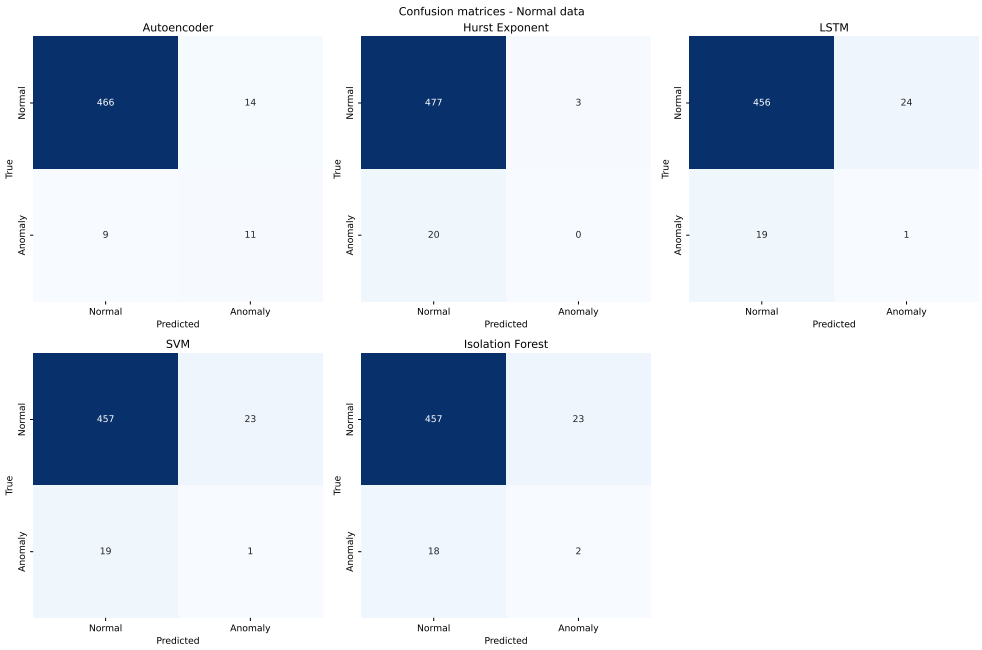

Figure 4.8.2 Confusion matrix of the algorithms performing on the unedited data

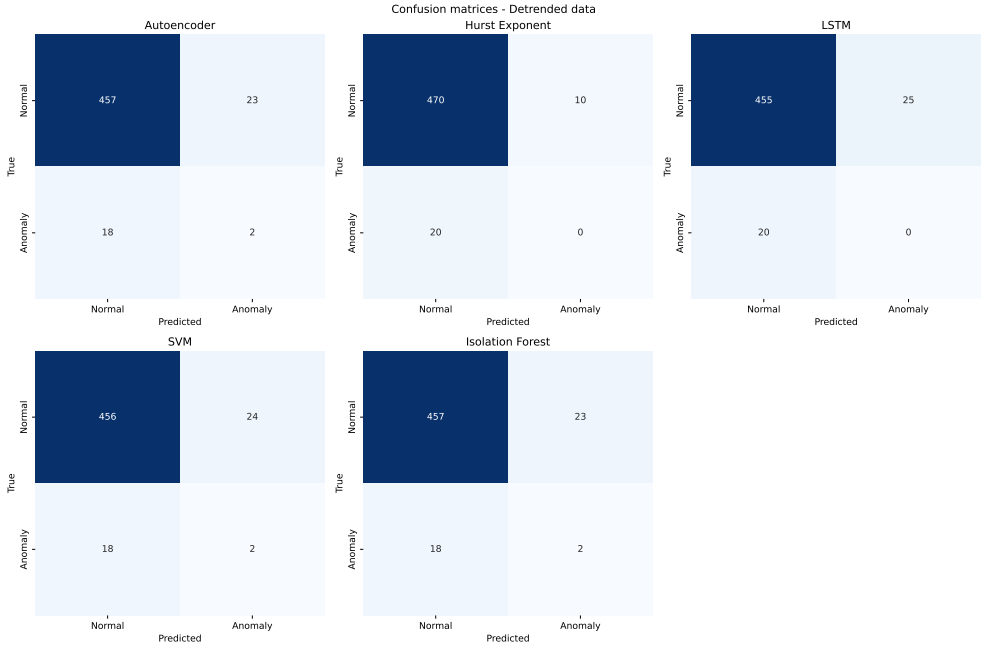

Figure 4.8.3 Confusion matrix of the algorithms performing on the detrended data

tion of the patterns and damaged the ability of the algorithms to find outliers. A possible cause for the disruption could be that the trend was very weak in the first place.

On the differenced data (see Figure 4.8.4), the SVM and Isolation Forest showed excellent performance, with the Autoencoder performing slightly worse than in the first case (Figure 4.8.2).

For the relative change in Figure 4.8.5, the detection has again worsened, except in the case of Autoencoder, which adapted very well.

When comparing these cases, what's surprising is that the LSTM and Hurst Exponential performed so badly. This, however, may be explained by the fact that the time series may contain some hidden patterns brought in during generating that influenced the capabilities of this algorithm to correctly predict the data. The most robust of these is certainly Autoencoder, which was capable of adapting to most of the data. The best-performing algorithms were Isolation Forest and SVM on differentiated data. This corresponds with the results of the previous section, where these two algorithms performed very well on differentiated data. The robustness of the Autoencoder also corresponds with the results of [1], even though we used different approaches to testing the artificial anomalies.

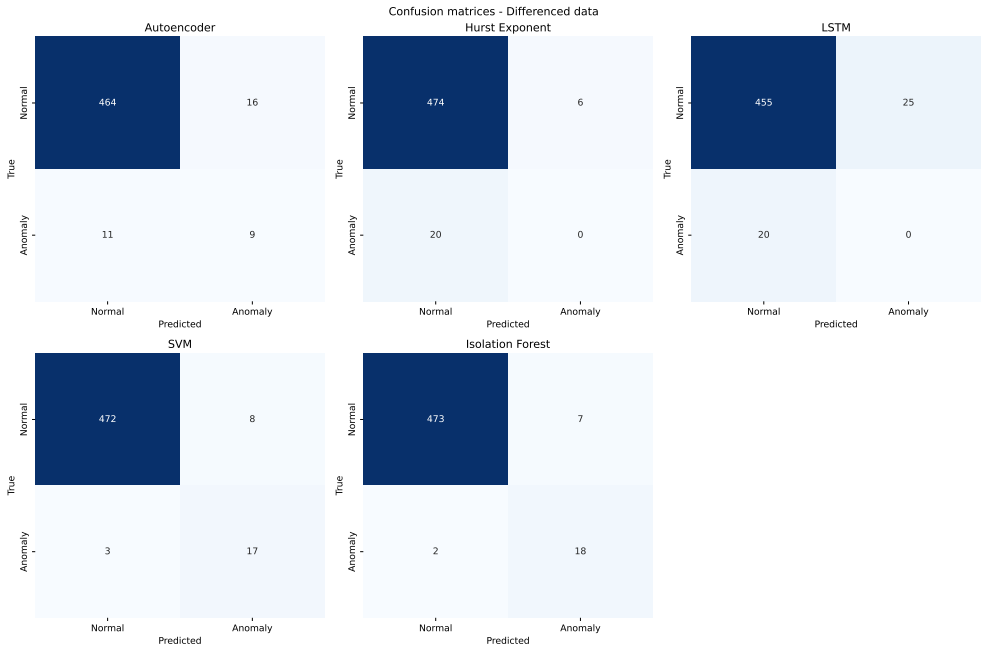

Figure 4.8.4 Confusion matrix of the algorithms performing on the differenced data

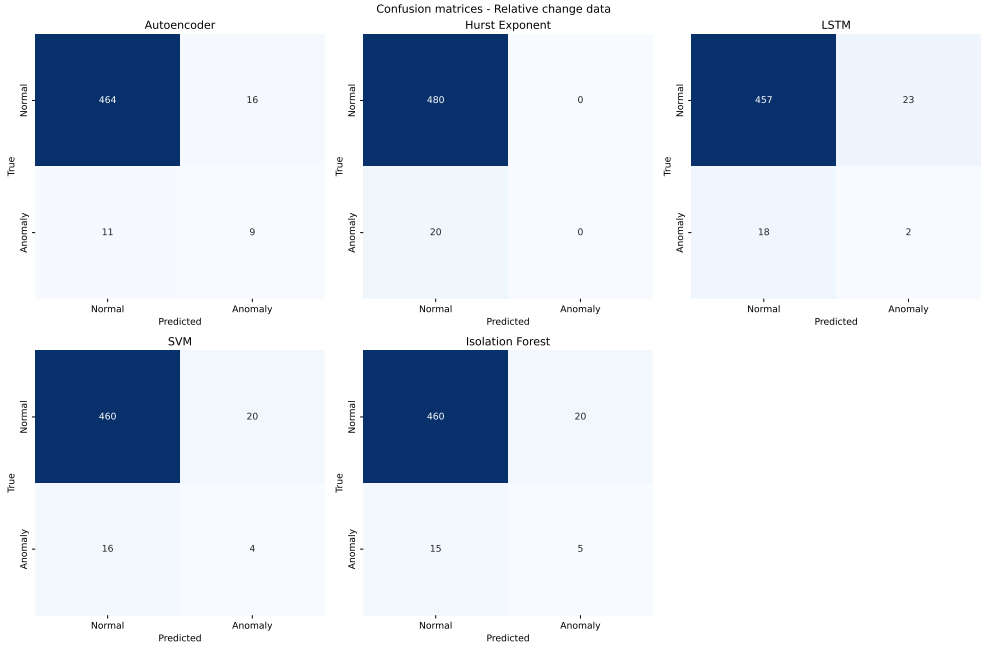

Figure 4.8.5 Confusion matrix of the algorithms performing on the data with relative change.

4.9 Anomaly detection overlap

Another important aspect we have to keep in mind when comparing the algorithms is the overlap in anomaly detections - which of the identified anomalies were also identified by other algorithms? Considering the overlap in anomaly detections among different algorithms helps us evaluate their consistency and reliability in various scenarios. This overlapping data can also provide indicators about how sure we can be that something is an anomaly when detected with the said algorithm.

The experiment has been done for all asset classes and works with the anomaly detections that have been discussed in the previous sections of this chapter (Sections 4.2, 4.3, 4.4, 4.5 and 4.6). Due to the similarity of the heatmaps inside the asset class, we will show just one heatmap for each asset class - Apple stocks for equities, EUR/USD FX rate for FX rates, and oil prices for commodities.

First are the stocks anomaly detection overlap heatmaps in Figure 4.9.1. The Apple heatmap demonstrates the similarity between the outputs of the algorithms run on differenced data, particularly the Isolation Forest, the SVM, and the Autoencoder. We can see particularly great accord between SVM and Isolated Forest detections. Also, what we can see is that the influence of the detrending in LSTM and Autoencoder is not as significant as it is for other methods. As for the Hurst exponent, unfortunately, the overlap is small with other methods, corresponding with its weak performance on these data.

For the EUR/USD/EUR FX rate in Figure 4.9.2, the resistance to the trend of the LSTM is more significant. We still have a strong overlap between Isolation Forest and SVM, as well as between the Autoencoder and the SVM. Hurst exponent now has better overlap with other methods, particularly with the isolation forest. Also, when we remove the trend, we can see that it brings Isolation Forests and Autoencoder together, which corresponds with the fact that their results were similar in these data.

Oil prices in Figure 4.9.3 continue to show good overlap between the SVM and Autoencoder, as well as the SVM and Isolation Forests. Interestingly, removing the trend did not affect almost any method, except for the Autoencoder, which confirms that the trend in this data is very weak. When we differentiate the data, we can see that the SVM and Isolation Forest overlap almost all samples.

In conclusion, we can confidently say that Autoencoder, SVM, and Isolation Forest show great overlap and signify that these methods give similar results on similar data. As for LSTM, we have confirmed its resistance to the trends. However, the results of this algorithm do not correspond very closely to the

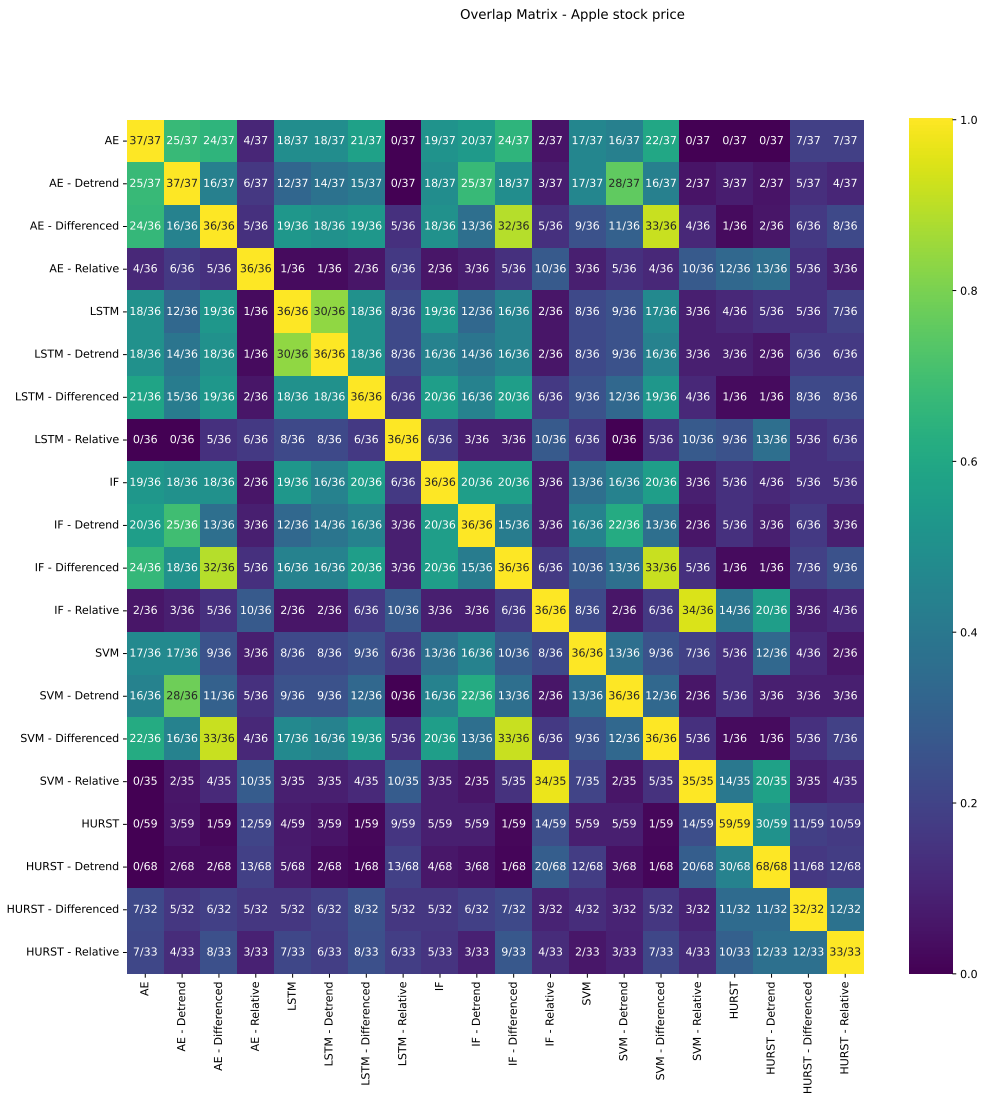

Figure 4.9.1 Overlap heatmap for Apple stock price anomaly detection.

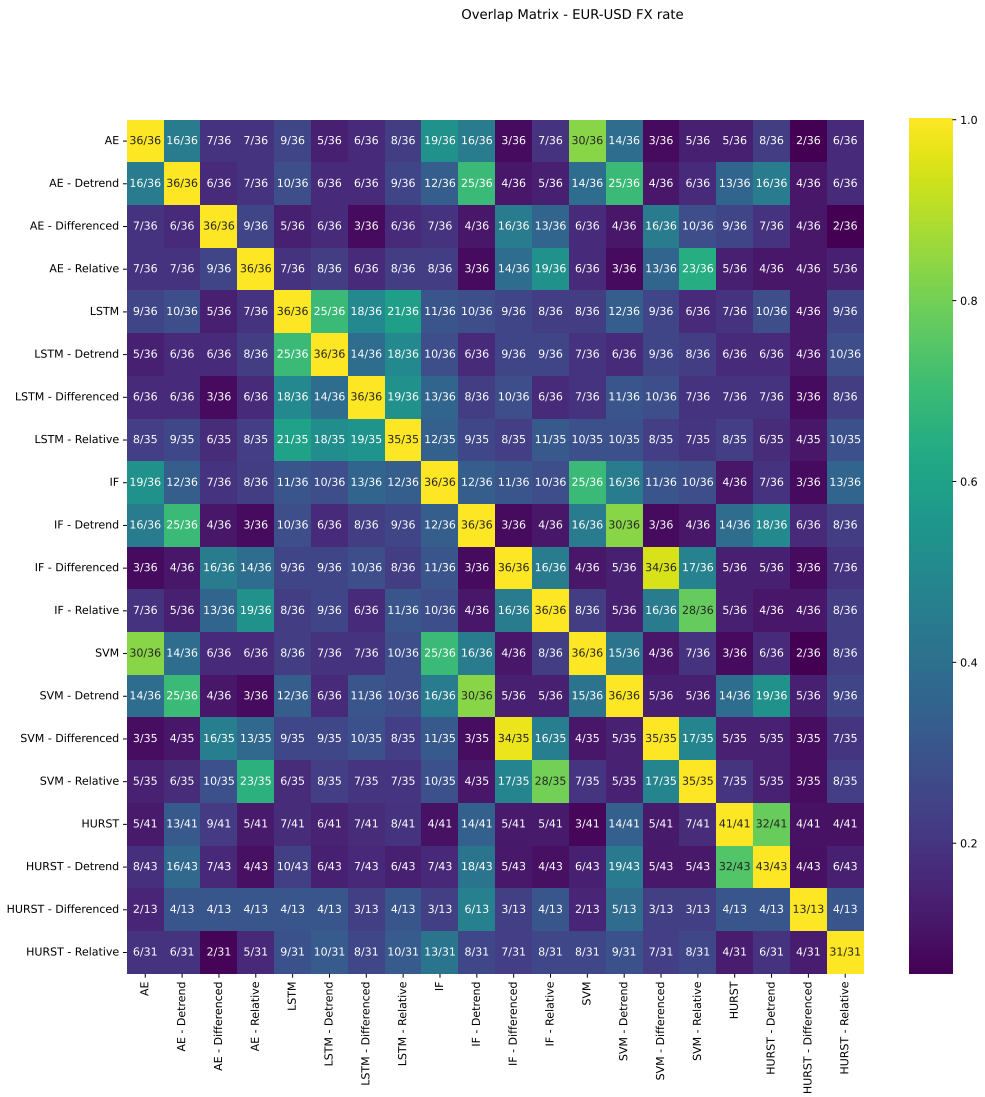

Figure 4.9.2 Overlap heatmap for EUR/USD FX rate anomaly detection.

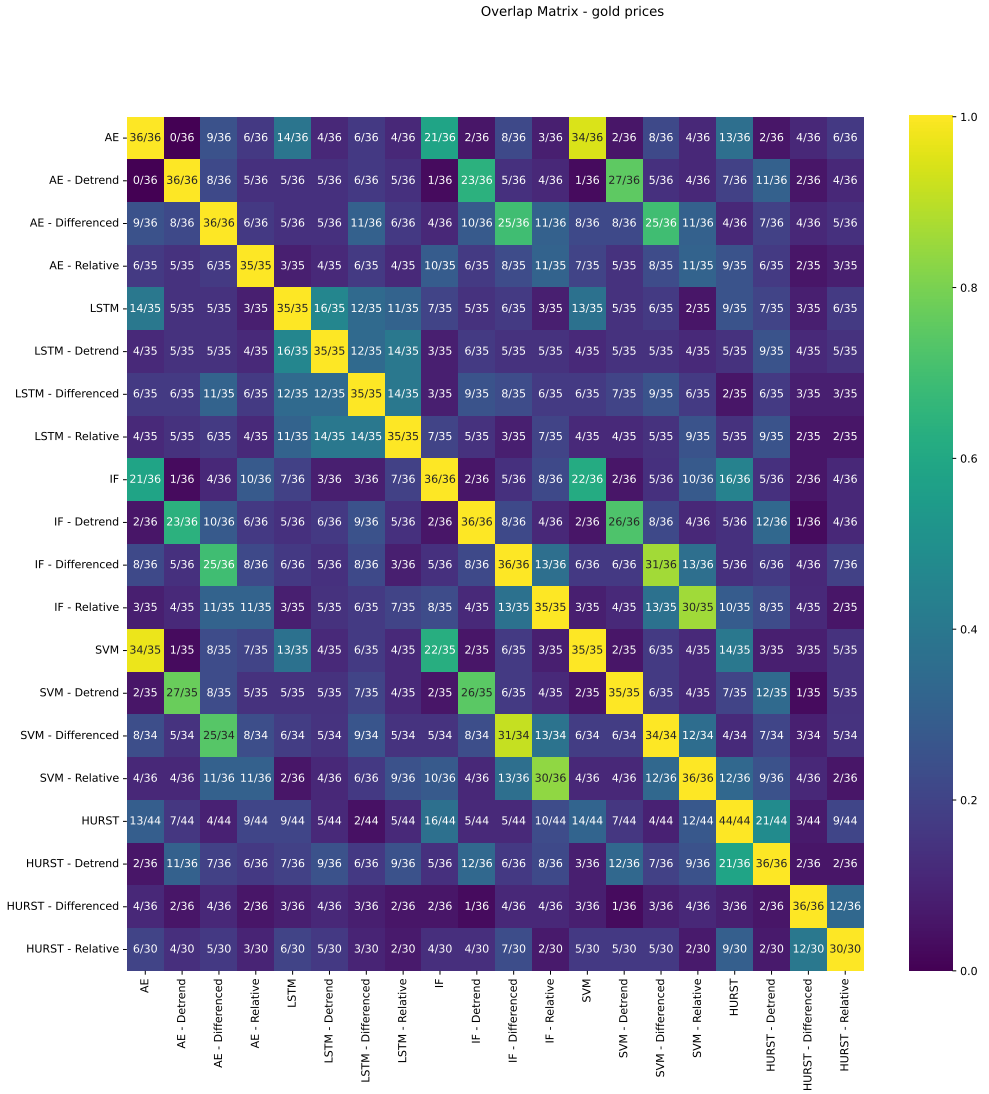

Figure 4.9.3 Overlap heatmap for oil price anomaly detection.

results of the other three. As for the Hurst Exponent, we have additional indices that tell us that it is not suitable for this task.

4.10 Silouhette scores

	Apple	EUR-JPY	EUR-USD	gold	oil	SP500
AE	0.76	0.41	0.31	0.4	0.3	0.64
AE - Detrend	0.39	0.43	0.32	0.34	0.33	0.46
AE - Differenced	0.68	0.44	0.42	0.42	0.43	0.52
AE - Relative	0.21	0.42	0.38	0.44	0.37	0.43
LSTM	0.56	0.17	0.1	0.16	0.14	0.27
LSTM - Detrend	0.22	0.17	0.04	0.09	0.14	0.14
LSTM - Differenced	0.52	0.23	0.12	0.12	0.03	0.33
LSTM - Relative	0.12	0.15	0.11	0.1	0.05	0.15
IF	0.55	0.4	0.44	0.36	0.38	0.4
IF - Detrend	0.32	0.45	0.43	0.49	0.37	0.27
IF - Differenced	0.67	0.51	0.54	0.49	0.49	0.59
IF - Relative	0.5	0.53	0.49	0.44	0.53	0.52
SVM	0.4	0.48	0.38	0.4	0.38	0.36
SVM - Detrend	0.35	0.48	0.43	0.43	0.37	0.31
SVM - Differenced	0.67	0.51	0.54	0.5	0.49	0.59
SVM - Relative	0.5	0.53	0.5	0.45	0.54	0.51
HURST	-0.14	-0.01	-0.01	0.21	0.18	-0.09
HURST - Detrend	-0.03	-0.01	0.17	0.22	0.16	0.12
HURST - Differenced	0.03	-0.17	0.21	-0.07	0.02	0.12
HURST - Relative	-0.06	-0.22	0.13	-0.08	-0.03	0.03

Table 4.1 Silhouette scores.

When comparing the silhouette scores in Table 4.1 that determine the level of separation, we can see that for Apple stocks, the best separation was achieved by the usage of Autoencoder, Isolation Forest, and LSTM for normal data, however for differenced data, while the Autoencoder, Isolation Forest still gave best results, the SVM was better than LSTM. For the S&P500, this is the same, even though the separation overall was not as great as for the Apple stocks. For the FX rates, the best choice was the SVM and Isolation Forest on differenced data, which is similar to the results observed in previous sections. The commodities were modeled better by the SVM, with differenced datasets, or the data with relative changes.

Chapter 5

Conclusion

The analysis done on anomaly detection algorithms in the previous chapter, where we applied them to various financial datasets, shows distinct performance characteristics for the Autoencoder, LSTM, and other models like Isolation Forest, SVM, and the Hurst Exponent. These results can give us insight into choosing the correct algorithm for our data.

The Autoencoder demonstrated high sensitivity to extreme values in datasets like Apple stocks and the S&P500, which mainly highlighted the most significant and lowest values. This characteristic makes the Autoencoder suitable for datasets with significant oscillations or where the primary interest is capturing the most significant anomalies. However, a strong trend can reduce its effectiveness, as seen with Apple's stocks and S&P500. In contrast, the Autoencoder's performance improved for data with a more stable mean or less pronounced trend, like FX rates and oil prices. As shown in the experiments, differencing the data before applying the Autoencoder generally enhances its ability to detect anomalies by reducing the distraction of a long-term trend.

The LSTM model, with its ability to leverage past information, was more sensitive to changes over time, making it highly suitable for financial time series where past patterns can inform future anomalies. Unlike the Autoencoder, LSTM was able to identify anomalies throughout the entire dataset, not just the extreme values or the latest trends, especially on the detrended data. This was particularly evident in detecting anomalies in stock price and FX rates, where LSTM captured market dynamics around significant dips and peaks.

For Autoencoder and LSTM, applying transformations like differencing and using relative change data sometimes helped highlight anomalies more evenly across the time series. These transformations made the algorithms less sensitive to absolute changes and more to relative changes, which is better in financial data analysis.

The Isolation Forest tended to identify anomalies in areas with sparse data or where data points were isolated from the central cluster. This characteristic makes it particularly useful for identifying outliers in a more scattered context, which can be beneficial in detecting unusual market conditions. However, its performance varied across different assets and adjustments, indicating that its effectiveness might depend on the specific structure and distribution of the data.

The findings regarding the LSTMs correspond to the findings of the previous research, where the LSTM was labeled as a preferred algorithm. However, this thesis also focused on adjusted data, where we found out that other algorithms sometimes outperformed the LSTM.

The two methods not described in the original research were the Hurst Exponent and the Support Vector Machine.

The Hurst Exponent, which measures the long-term memory of a time series, showed limited capabilities. It focused on macroscopic trends rather than the anomalies we tried to identify in this research. This suggests that while the Hurst Exponent can help identify longer market trends, it could be more effective for identifying specific anomalous events.

The Support Vector Machine (SVM) performs very well on detrended and differenced data, effectively highlighting anomalies by focusing on deviations and significant changes. The SVM did exceptionally well on the FX rates and oil prices datasets.

Unfortunately, the retraining strategies could not be replicated because the data used for it in the original research was not available. With the available data, the tested algorithm could not reconstruct the data well enough.

As for the artificial anomalies, the Isolated Forest and SVM outperformed other algorithms when talking about differenced data. For the unadjusted data, the Autoencoder was the most efficient algorithm. The author of this thesis used a different approach than the authors of the original research; however, the conclusion - that the Autoencoder is the best model for this task is proper, although the original research did not mention differentiating the data, in which case other mentioned algorithms were better.

Heatmaps and silhouette scores provided additional insights into the clustering and distribution of anomalies. Heatmaps were very useful in visualizing the places where models were more consistent in the identified anomalies, highlighting the overlaps across the entire dataset. Silhouette scores, on the other hand, helped us to see the quality of the separation of identified anomalies, giving a quantitative measure to the visual measure from heatmaps.

From the findings, the author of this thesis formulated the following recommendations for the usage of this algorithm in real-life scenarios:

- For data with significant trends or where the interest is in capturing the most pronounced anomalies, detrending and using the Autoencoder can be effective.
- For time series where past information predicts future anomalies, the LSTM model is recommended due to its ability to utilize sequential data.
- In cases where anomalies are not linked to the magnitude of changes but to their isolation from the other points, the Isolation Forest, or the SVM, is the best choice.
- The Isolation Forest and SVM are both very useful in predicting stocks like Apple or S&P500, particularly with relative change data.

Overall, no algorithm outperformed all others on the data, and the choice of the algorithm shows to be linked to which data - asset class in this case - we are currently processing. It is recommended that multiple algorithms be used at once to utilize their unique abilities to the fullest.

Bibliography

1. BIANCHETTI, Marco; SCARINGI, Marco. Learning Market Data Anomalies. In: *Proceedings of QuantMinds International*. London, 2023. Electronically approved by Ing. Magda Friedjungová, Ph.D. on 22 February 2024 in Prague.
2. PARIMALA, Venkata Krishna. Introductory Chapter: Anomaly Detection – Recent Advances, AI and ML Perspectives and Applications. In: PARIMALA, Venkata Krishna (ed.). *Anomaly Detection*. Rijeka: IntechOpen, 2024, chap. 1. Available from DOI: 10.5772/intechopen.113968.
3. JI, Il Hwan; LEE, Ju Hyeon; KANG, Min Ji; PARK, Woo Jin; JEON, Seung Ho; SEO, Jung Taek. Artificial Intelligence-Based Anomaly Detection Technology over Encrypted Traffic: A Systematic Literature Review. *Sensors*. 2024, vol. 24, no. 3. ISSN 1424-8220. Available from DOI: 10.3390/s24030898.
4. THUDUMU, Srikanth; BRANCH, Philip; JIN, Jiong; SINGH, Jugdutt (Jack). A comprehensive survey of anomaly detection techniques for high dimensional big data. *Journal of Big Data*. 2020, vol. 7, no. 1, p. 42. Available from DOI: 10.1186/s40537-020-00320-x.
5. DOMB, Menachem; JOSHI, Sujata; KHN, Arulmozhi. Anomaly Detection in IoT: Recent Advances, AI and ML Perspectives and Applications. In: PARIMALA, Venkata Krishna (ed.). *Anomaly Detection*. Rijeka: IntechOpen, 2023, chap. 3. Available from DOI: 10.5772/intechopen.111944.
6. ENDERLEIN, G. Hawkins, D. M.: Identification of Outliers. Chapman and Hall, London – New York 1980, 188 S., £ 14, 50. *Biometrical Journal*. 1987, vol. 29, no. 2, pp. 198–198. Available from DOI: https://doi.org/10.1002/bimj.4710290215.

7. GRUBBS, Frank E. Procedures for Detecting Outlying Observations in Samples. *Technometrics*. 1969, vol. 11, no. 1, pp. 1–21. Available from DOI: 10.1080/00401706.1969.10490657.

8. MALHOTRA, Pankaj; RAMAKRISHNAN, Anusha; ANAND, Gaurangi; VIG, Lovekesh; AGARWAL, Puneet; SHROFF, Gautam. Anomaly Detection in Time-series: An Approach Using LSTM and Autoencoders. *Journal of Systems and Software*. 2016, vol. 117, pp. 496–506.

9. MALHOTRA, Pankaj; RAMAKRISHNAN, Anusha; VIG, Lovekesh; AGARWAL, Puneet; SHROFF, Gautam. Enhancing Time-Series Anomaly Detection Using Recurrent Neural Networks. *ArXiv*. 2017, vol. abs/1707.01780.

10. WANG, Zexin; PEI, Changhua; MA, Minghua; WANG, Xin; LI, Zhihan; PEI, Dan; RAJMOHAN, Saravan; ZHANG, Dongmei; LIN, Qingwei; ZHANG, Haiming; LI, Jianhui; XIE, Gaogang. Revisiting VAE for Unsupervised Time Series Anomaly Detection: A Frequency Perspective. *arXiv preprint arXiv:2302.00433*. 2024.

11. MUNIR, Mohsin; SIDDIQUI, Shoaib Ahmed; DENGEL, Andreas R.; AHMED, Sheraz. DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series. *IEEE Access*. 2019, vol. 7, pp. 1991–2005. Available also from: https://api.semanticscholar.org/CorpusID:57756774.

12. MUNIR, Mohsin; SIDDIQUI, Shoaib Ahmed; CHATTHA, Muhammad Ali; DENGEL, Andreas R.; AHMED, Sheraz. FuseAD: Unsupervised Anomaly Detection in Streaming Sensors Data by Fusing Statistical and Deep Learning Models. *Sensors (Basel, Switzerland)*. 2019, vol. 19. Available also from: https://api.semanticscholar.org/CorpusID:171093087.

13. ROSNER, Bernard. Seasonal Hybrid Extreme Studentized Deviate Test for Time-Series Anomaly Detection. In: *KDD*. 1983.

14. SCHÖLKOPF, Bernhard; WILLIAMSON, Robert; SMOLA, Alex; SHAWE-TAYLOR, John; PLATT, John. Support Vector Method for Novelty Detection. In: 1999, vol. 12, pp. 582–588.

15. LIU, Fei Tony; TING, Kai Ming; ZHOU, Zhi-Hua. Isolation Forest. In: *2008 Eighth IEEE International Conference on Data Mining*. 2008, pp. 413–422. Available from DOI: 10.1109/ICDM.2008.17.

16. YEPMO, Véronne; SMITS, Grégory; LESOT, Marie-Jeanne; PIVERT, Olivier. Leveraging an Isolation Forest to Anomaly Detection and Data Clustering. *Data & Knowledge Engineering*. 2024, vol. 151, p. 102302. ISSN 0169-023X. Available from DOI: https://doi.org/10.1016/j.datak.2024.102302.

17. NAYEBI, Armin; TIPIRNENI, Shalini; FOREMAN, Brandon; RATCLIFF, John; REDDY, Chandan K.; SUBBIAN, Vignesh. Recurrent Neural Network based Time-Series Modeling for Long-term Prognosis Following Acute Traumatic Brain Injury. *AMIA Annual Symposium Proceedings*. 2022, vol. 2021, pp. 900–909.

18. YIN, Wenpeng; KANN, Katharina; YU, Mo; SCHÜTZE, Hinrich. *Comparative Study of CNN and RNN for Natural Language Processing*. 2017. Available from arXiv: 1702.01923 [cs.CL].

19. GRAVES, Alex; MOHAMED, Abdel-rahman; HINTON, Geoffrey. *Speech Recognition with Deep Recurrent Neural Networks*. 2013. Available from arXiv: 1303.5778 [cs.NE].

20. HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. *Neural computation*. 1997, vol. 9, no. 8, pp. 1735–1780.

21. OROJO, Oluwatamilore; TEPPER, Jonathan; MCGINNITY, T.M.; MAHMUD, Mufti. The Multi-Recurrent Neural Network for State-Of-The-Art Time-Series Processing. *Procedia Computer Science*. 2023, vol. 222, pp. 488–498. ISSN 1877-0509. Available from DOI: https://doi.org/10.1016/j.procs.2023.08.187. International Neural Network Society Workshop on Deep Learning Innovations and Applications (INNS DLIA 2023).

22. MICHELUCCI, Umberto. *An Introduction to Autoencoders*. 2022. Available from arXiv: 2201.03898 [cs.LG].

23. RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning internal representations by error propagation. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362. ISBN 026268053X.

24. BANK, Dor; KOENIGSTEIN, Noam; GIRYES, Raja. *Autoencoders*. 2021. Available from arXiv: 2003.05991 [cs.LG].

25. PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011, vol. 12, pp. 2825–2830.

26. MANDELBROT, Benoit B; WALLIS, James R. Robustness of the rescaled range R/S in the measurement of noncyclic long run statistical dependence. *Water resources research*. 1969, vol. 5, no. 5, pp. 967–988.

27. GRECH, D; MAZUR, Z. Can one make any crash prediction in finance using the local Hurst exponent idea? *Physica A: Statistical Mechanics and its Applications*. 2004, vol. 336, no. 1, pp. 133–145. ISSN 0378-4371. Available from DOI: https://doi.org/10.1016/j.physa.2004.01.018. Proceedings of the XVIII Max Born Symposium "Statistical Physics outside Physics".

28. IVANOV, Plamen Ch; AMARAL, Luis A Nunes; GOLDBERGER, Ary L; HAVLIN, Shlomo; ROSENBLUM, Michael G; STRUZIK, Zbigniew R; STANLEY, H Eugene. Multifractality in human heartbeat dynamics. *Nature*. 1999, vol. 399, no. 6735, pp. 461–465.

29. CHANDRASEKARAN, Sivapragasam; POOMALAI, Saravanan; SAMINATHAN, Balamurali; SUTHANTHIRAVEL, Sumila; SUNDARAM, Keerthi; ABDUL HAKKIM, Farjana Farveen. An investigation on the relationship between the Hurst exponent and the predictability of a rainfall time series. *Meteorological Applications*. 2019, vol. 26, no. 3, pp. 511–519. Available from DOI: https://doi.org/10.1002/met.1784.

30. PENG, C.-K.; BULDYREV, S. V.; HAVLIN, S.; SIMONS, M.; STANLEY, H. E.; GOLDBERGER, A. L. Mosaic organization of DNA nucleotides. *Phys. Rev. E*. 1994, vol. 49, pp. 1685–1689. Available from DOI: 10.1103/PhysRevE.49.1685.

31. MOTTL, Matthias. *hurst: Estimation of the Hurst exponent for long time series*. 2017. Version 1.0.0. Available also from: https://github.com/Mottl/hurst.

32. ROUSSEEUW, Peter J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*. 1987, vol. 20, pp. 53–65. ISSN 0377-0427. Available from DOI: https://doi.org/10.1016/0377-0427(87)90125-7.

33. CERQUITELLI, Tania; GARZA, Paolo. Distributed Silhouette Algorithm: Evaluating Clustering on Big Data. *arXiv preprint arXiv:2303.14102*. 2023.

34. PETROVIC, Slobodan V. A Comparison Between the Silhouette Index and the Davies-Bouldin Index in Labelling IDS Clusters. In: 2006. Available also from: https://api.semanticscholar.org/CorpusID:14657641.

35. CALIŃSKI, Tadeusz; HARABASZ, Joachim. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*. 1974, vol. 3, pp. 1–27. Available also from: https://api.semanticscholar.org/CorpusID:122217223.

36. JOLLIFFE, Ian T.; CADIMA, Jorge. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 2016, vol. 374, no. 2065, p. 20150202. Available from DOI: 10.1098/rsta.2015.0202.

37. MICHELUCCI, Umberto. *An Introduction to Autoencoders*. 2022. Available from arXiv: 2201.03898 [cs.LG].

38. TORABI, Hasan; MIRTAHERI, Seyedeh Leili; GRECO, Sergio. Practical autoencoder based anomaly detection by using vector reconstruction error. *Cybersecurity*. 2023, vol. 6, no. 1, p. 1. Available from DOI: 10.1186/s42400-022-00134-9.

39. ARAZO, Eric; ORTEGO, Diego; ALBERT, Paul; O'CONNOR, Noel E.; MCGUINNESS, Kevin. *Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning*. 2020. Available from arXiv: 1908.02983 [cs.CV].

40. BRADLEY, Andrew P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*. 1997, vol. 30, no. 7, pp. 1145–1159. ISSN 0031-3203. Available from DOI: https://doi.org/10.1016/S0031-3203(96)00142-2.

41. SERVICES, Harvard Business. *How Exchange Rates Affect Your Business* [online]. Harvard Business Services, 2022. [visited on 2024-05-05]. Available from: https://www.delawareinc.com/blog/exchange-rates/. Accessed on: May 5, 2024.

42. CAPORALE, G. M.; PLASTUN, Alex. Daily abnormal price changes and trading strategies in the FOREX. *Journal of Economic Studies*. 2021, vol. 48, no. 1, pp. 211–222. Available from DOI: 10.1108/JES-11-2019-0503.

43. KICHIKAWA, Yuichi; IYETOMI, Hiroshi; AOYAMA, Hideaki; FUJIWARA, Yoshi; YOSHIKAWA, Hiroshi. Interindustry linkages of prices-Analysis of Japan's deflation. *PLoS One*. 2020, vol. 15, no. 2, e0228026. Available from DOI: 10.1371/journal.pone.0228026.

44. CONSTÂNCIO, V. The Return of Fiscal Policy and the Euro Area Fiscal Rule. *Comp Econ Stud*. 2020, vol. 62, no. 3, pp. 358–372. Available from DOI: 10.1057/s41294-020-00122-3. Epub 2020 Jul 6.

45. BAGCHI, Bhaskar; PAUL, Biswajit. Effects of Crude Oil Price Shocks on Stock Markets and Currency Exchange Rates in the Context of Russia-Ukraine Conflict: Evidence from G7 Countries. *Journal of Risk and Financial Management*. 2023, vol. 16, no. 2. ISSN 1911-8074. Available from DOI: 10.3390/jrfm16020064.

Enclosed medium contents

- `readme.txt` ..... description of the content
- `src`
  - `impl` ..... Source codes of implementation
  - `thesis` ..... thesis source code in form of LaTeX
- `text` ..... text of the thesis
  - `thesis.pdf` ..... text of the thesis in the form of a PDF file