National Research University Higher School of Economics
Faculty of Computer Science
Programme 'Master of Data Science'

MASTER'S THESIS

Comparison of Financial Time series methods for Anomaly Detection in Individual Stocks

Student: James Brian Sullivan

Supervisor: Bachelor's Degree, Data Scientist Sber, Andrey Chankin

Moscow, 2021

Abstract

An important research field for some time, anomaly detection on time series data, has recently seen many machine learning approaches added to existing statistical techniques. Although there have been many studies of anomaly detection of stock market prices, most have used daily intervals and univariate approaches. This thesis tries to fill in the lack of comparisons of anomaly detection methods in more challenging intraday price interval stock prices and the paucity of comparisons, including more recent methods. Given the unsupervised nature of stock price anomalies, we first try to detect all three types of (global, conditional, and collective) outliers in artificially constructed situations. If successful, we then try to detect outliers injected into the actual stock prices of several Nikkei 225 constituents using minute intervals over two years. We hope to provide insight into which methods are easily suitable for anomaly detection in intraday stock prices.

Table of Contents

Definitions, Designations, and Abbreviations

ABOD Angle-based Outlier Detection
AC autocorrelation
ACF autocorrelation function
ADF augmented Dickey-Fuller
AIC Akaike information criterion
AR Autoregressive
ARIMA Autoregressive Integrated Moving Average
CBLOF Clustering-Based Local Outlier
Deep SVDD Deep Support Vector Data Description
EMH efficient market hypothesis
IFOREST Isolation Forest
i.i.d. independent and identically distributed
KNN k-Nearest Neighbors
KPSS Kwiatkowski-Phillips-Schmidt-Shin
LB Ljung-Box
LOF Local Outlier Factor
LRD Local Reachability Density
MA Moving Average
MAE mean absolute error
NN Neural Network
PACF partial autocorrelation function
PCA Principal Component Analysis
RMSE root mean square error
SD standard deviation
SIC Schwartz information criterion
SVM Support Vector Machines
OC-SVM One Class SVM

Introduction

Anomaly detection is a critical task for financial market participants and regulatory authorities. Although there have been many studies of anomaly detection of stock market prices, most have used daily intervals and univariate approaches. There are valid reasons for academics preferring daily intervals for research, such as the data being publicly available and more readily usable without preprocessing. Intraday data often requires costly licenses making replication and verification of research results more difficult, and the non-continuous nature of stock exchanges means tiresome and error-prone preprocessing of the data is necessary. However, market participants want the information as quickly as possible, real-time tick-by-tick or order book-based anomaly detection with the attendant complications. We choose minute intervals as a reasonable middle ground; minute-by-minute pricing includes intraday considerations to make the preprocessing and modeling more than a toy exercise while at the same time avoiding the excessive complication that the extreme volume and velocity that tick data would bring. This thesis tries to fill in the lack of comparisons of anomaly detection methods for the more challenging intraday price interval stock prices, including more recent methods. Anomalies mean different things in different situations, so we will look at this from the viewpoint of providing information for a human to use.

In stock price time series, “normal” or “outlier” labeled results for training and validation are generally not available, and an **unsupervised learning** method has to be used. Unsupervised anomaly detection methods implicitly assume that the normal objects are somewhat “clustered.” Given the unsupervised nature of stock price anomalies, we first try to detect all three types of (global, conditional, and collective) outliers in entirely artificially constructed situations. If successful, we then try to detect outliers injected into the actual stock prices of Nikkei 225 constituents using minute intervals over two years. We hope to provide insight into which methods are suitable for anomaly detection in intraday stock prices and why by reviewing the results.

Overview

1. Basics: Review the fundamental concepts of anomaly detection with time series and the three types of outliers. Terms are defined to avoid ambiguity and confusion.
2. Selected Anomaly detection approaches: This chapter briefly explains four different categories of anomaly detection algorithms,
3. Initial Investigation of Data: Details of data source and preprocessing.
4. Methodology: Details around the setup of anomaly detection methods,
5. Results: This chapter lays out the results.
6. Conclusion: A summary of how to interpret the results and areas for future work.

Basics

Definition of Anomaly

In the context of stock prices, there are various definitions of the term anomaly. These definitions are usually based on an objective, such as finding opportunities based on deviations of the Efficient Market Hypothesis EMH or detection of insider trading. However, as there is no widely accepted mathematical definition of anomaly, in this thesis, we use anomaly and outlier interchangeably in a more generic fashion as put forth by D. Hawkins more than 50 years ago

Definition: An anomaly is an observation or a sequence of observations *that deviates significantly from the rest of the data, as if it were generated by a different mechanism*.[1]

By definition, the set of anomalies forms a tiny part of the dataset, meaning the dataset will be very imbalanced. Imbalanced classifications pose a challenge for machine learning classification algorithms as these algorithms often assume an equal number of examples for each class.[2]

Note that the definition of anomaly differs from noise, which may also deviate significantly from the rest of the data but is not generated by a different mechanism that we are interested in and is instead a random error or variance in a measured variable. To distinguish between anomalies and noise, we need to justify why the anomalies detected are generated by some other mechanisms. This differentiation can be achieved by making various assumptions on the rest of the data and showing that the outliers detected violate those assumptions significantly. We should also distinguish between anomalies and novelties. Novelties are new different data points that have not been encountered before but are only considered anomalies the first time encountered, and after that are considered normal. For example, if stock prices jump up considerably, the first occurrence of the novel higher price should be considered an anomaly but not the successive prices at the same level.

Types of Anomalies

Anomalies are usually classified into three categories: point outliers, contextual (or conditional) outliers, and collective outliers.[3]

Global Outlier

In a given data set, a data point is a global outlier (also called a point anomaly) if it deviates significantly from the rest. Point anomalies are the most straightforward and most readily apparent type of outliers, and a stock price much higher or lower than any of the others would be an example. Most anomaly detection methods are for discovering point anomalies via an appropriate deviation measure with respect to all of the data being examined. These different ways of measuring deviation can then be used to categorize the different types of point anomaly detection methods.

Contextual Outliers

A data point is a contextual (also called conditional) outlier if it deviates significantly with respect to a specific context of the data point. In contextual outlier detection, the context must be specified as part of formulating the problem or objective. Contextual outliers are a generalization of a **local outlier**--a point whose density significantly deviates from the local area in which it occurs. The local area context is often temporal for time series data because time series data are records of a specific quantity over time. Stock prices typically decline after the ex-dividend date, so if they rose even moderately at that time, it might be a contextual outlier.

In contextual outlier detection, the attributes of the data are commonly divided into two types:

Contextual attributes: The contextual attributes of a data object define the object's context. In a stock example, the contextual attributes could include recent preceding prices, the ex-dividend date, Special Quotations date, the particular industry's performance, and the overall market's performance.

Behavioral attributes: These attributes define the object's characteristics. We use attributes to evaluate whether the object is an outlier in the context to which it belongs. In an example for stocks, the security price would be a behavioral attribute.

Collective Outliers

A subset of data points forms a collective outlier if the points as a whole deviate significantly from the entire data set, even if each data point taken individually would not be considered an outlier. A small group of closely clustered points would be a classic example when data points are otherwise spread out. The classic example given for time-series price data is that if the price of a well-traded stock remained at the same price for an extended period, which would be unusual for a liquid stock, then the individual points would not be outliers, but the group would be a **collective outlier**.

Time-series

Time series is an overloaded word that's meaning is dependent on context. More generally, a time series is a sequence of data points in successive order over a period of time, in contrast with cross-sectional data, which captures many different subjects at one point in time.

In academic literature about time series, the following more precise definitions are common[4]:

Sequential Data:

The order of the data matters, but the timestamp is irrelevant, or it does not matter.

Temporal Sequence:

In addition to preserving the order of data, each point is time-stamped at regular or irregular time intervals.

Time Series:

Ordered real-value measurements at regular time intervals. More formally, it is a realization of a particular stochastic process:

$$T = (t_0^d, t_1^d, ..., t_t^d), d \in \mathbb{N}_+, t \in \mathbb{N}$$

where *d* is the dimension of the time series.

Almost all public exchange-traded stock securities data is temporal sequence data as public exchanges are not usually open 24 hours a day, 365 days a year. However, they can be considered time series data during market hours in the more strict sense of the term.

Temporal continuity refers to the fact that patterns in the data are not expected to change significantly over time unless they are generated by a different mechanism than usual. Temporal continuity, a specific type of contextual locality, is usually robust within market hours and much weaker between closes and opens.

Stationarity

A common assumption in many algorithms and methods dealing with time series techniques is that the data are stationary.

Simply put, stationarity is when the statistical properties of a process generating a time series do not change over time. The time series can change over time, but how it changes over time cannot change. More formally,

X_t is a stationary time series if the distribution of $(x_t, \ldots, x_{t+s})$ is equal for all s.

A stationary process's mean, variance, and autocorrelation structure do not change over time.

Realistically, price time series data is not stationary, especially over extended periods. This difference will necessitate data transformations and relaxation of the strictness of the assumption stationarity in practice.

Data streams

Time series data are received in a batch (offline) or streaming (online) fashion. The entire stream may be available for analysis in the offline case, whereas only the stream up to the current time is available in the online case. The advantage of seeing succeeding values makes it easier to discover outliers using offline data, but for many practical use cases, anomalies need to be discovered from streaming data without the help of future data points.

The literature notes the following properties of data streams as special challenges for detecting anomalies.[5]

Transience Data points in a data stream are naturally temporary, with the most interest in the most recent value, particularly for streams of stock prices.

Unending The entire data stream is never available and never ends, so the detection method must be incremental, storing either a limited amount of relevant data or a summary of the data.

Arrival Rate The arrival rate may be variable (temporal data) or fixed (the more strict definition of Time Series). We expect outlier detection methods to process the last data point before the next one arrives.

Concept Drift This term refers to the underlying probability distribution of a data stream changing over time. The term is particularly relevant in streaming data for price data of financial instruments; a given probability distribution will cause market participants to act in a particular fashion to gain profits, which will, in turn, change the underlying probability distribution.

These challenges are practical concerns for any real-time price anomaly detection system but are out of the scope of this paper. All of the methods considered are practical for use at a minute time scale, but most would be impractical for tick data.

Sliding Window

The **sliding window** concept is often used for streaming data to limit the amount of information to be considered at a given point in time. The window is identified by starting and ending points. These points are moved in the same direction and shifted the same number of units. Letting W be the window size, only the last W records to arrive in a data stream are relevant at any point in time. Usually, a window's data points overlap between the next window and the previous window. We use a sliding window for the 10-day volatility calculation to help with detecting collective anomalies.

Anomaly Detection Methods

One way of classifying machine learning methods is to divide them into two main types: **model methods** and **instance-based learning methods**. Model methods, which are primary statistical methods in the case of anomaly detection, use the data to create a model and then score each point based on its deviation from that model. Instead of using explicit generalization to create a model, instance-based learning methods compare new problem instances to the entire dataset. There are numerous ways model methods can be further broken down, such as traditional statistical and neural network methods. Instance-based learning methods for outlier detection are often further classified into proximity-based and clustering-based methods.

Model Methods

Traditional Statistical

These methods assume that a selected statistical stochastic model generates regular data points and that data not likely to be generated by the selected probability distribution are outliers. It follows that the effectiveness of statistical methods greatly depends on whether the assumptions made for the statistical model are correct for the given data. The distribution of asset returns is well-studied and among the most critical stylized facts are that asset returns have heavier tails than the gaussian distribution and the distribution of asset returns seems to be more peaked. Accordingly, the normal distribution may not be appropriate for any statistical method modeling stock prices. Statistical methods include both parametric methods, which assume that the regular data points are generated by a parametric distribution with parameter Θ (such as Mahalanobis distance or χ^2-statistic) and nonparametric methods (such as histogram and kernel density estimation), which do not assume an a priori statistical model.[2] Statistical methods can handle both univariate and multivariate (such as Mahalanobis distance) but are often tricky for high-dimensional data, especially streaming data.

Classification based

Classification methods train a classifier to separate areas containing normal data and other areas. Learning a good feature space for performing such separation is one of the challenges. Commonly used classification methods include One-class SVM and Isolation Forests.

Dimensionality Reduction

These methods transform data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains most of the meaningful properties of the original data. Any data not easily reconstructed from the lower-dimensional representation is unusual and can be considered an anomaly. Principal Component Analysis is a typical example.

Neural Network

Neural networks are collections of nodes forming probability-weighted associations between inputs and outputs stored within the net's data structure. Depending on the definition of a statistical model and which machine learning models are considered neural networks, these networks can be considered another statistical method. However, given their widespread use, effectiveness, and many variants, neural networks warrant their own Deep Anomaly Detection category.[6]

Instance-based Learning Methods

Proximity-based

Proximity-based methods straightforwardly assume that in higher dimensional space, normal data points are close, and any points significantly farther away from their neighbors are anomalies. The two main types of proximity-based outlier detection are distance-based *DB(r, π)* and density-based outlier detection. A *DB(r, π)* detection method considers the neighborhood of a point, which is defined by a given radius *r*. A point is then considered a distance-based outlier if its neighborhood does not include a sufficient number of other points. A density-based outlier detection method, such as Local Outlier Factor (LOF), investigates the density of a point and that of its neighbors. In density-based outlier detection, a point is identified as an outlier if its density is relatively much lower than its neighbors. Relative density is the key to finding local outliers; we use the parameter *k* to quantify the neighborhood and do not need to specify the minimum number of objects in the neighborhood as a density requirement instead of calculating the local reachability density for a point.

Clustering-based

Methods such as DBSCAN assume that normal data points belong to large and dense clusters, whereas outliers belong to small or sparse clusters or do not belong to any clusters. In density-based clustering, to determine whether an object can be considered a core object in a density-based cluster, two parameters are used: a radius parameter, *r*, to specify the range of the neighborhood and the minimum number of points in the *r*-neighborhood. Both parameters are global and applied to every object.

Angle-based

These methods of detecting outliers use the fact that the angle from one point to two other points will usually be larger if they are close together in a cluster and smaller if they are far apart. Measures of angle can be far more efficient when a data set's dimensions grow larger. [7]

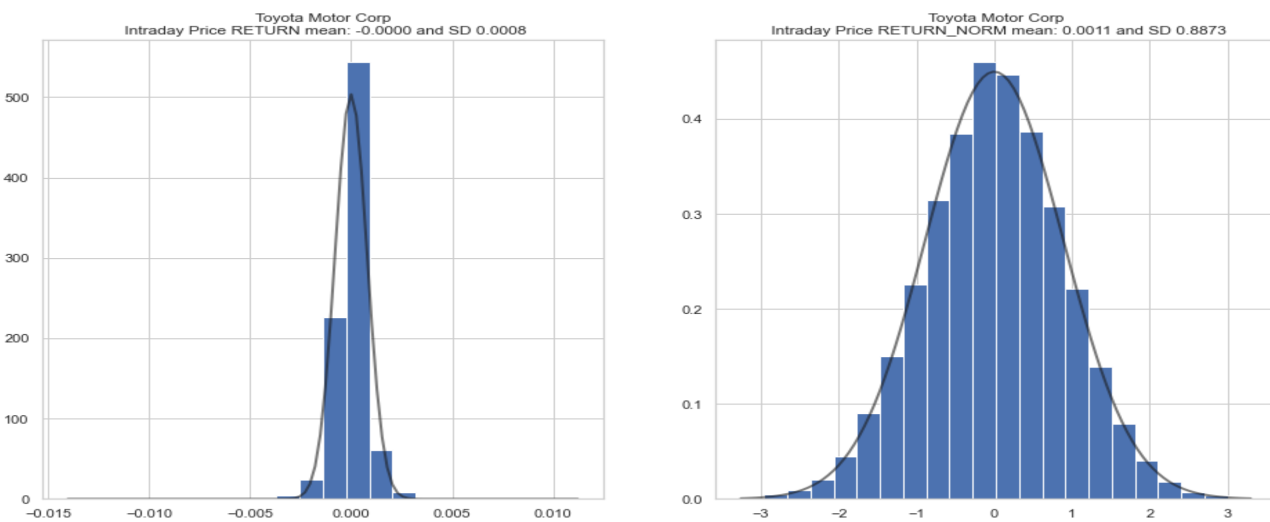
Initial Investigation of Price Data

Data Source

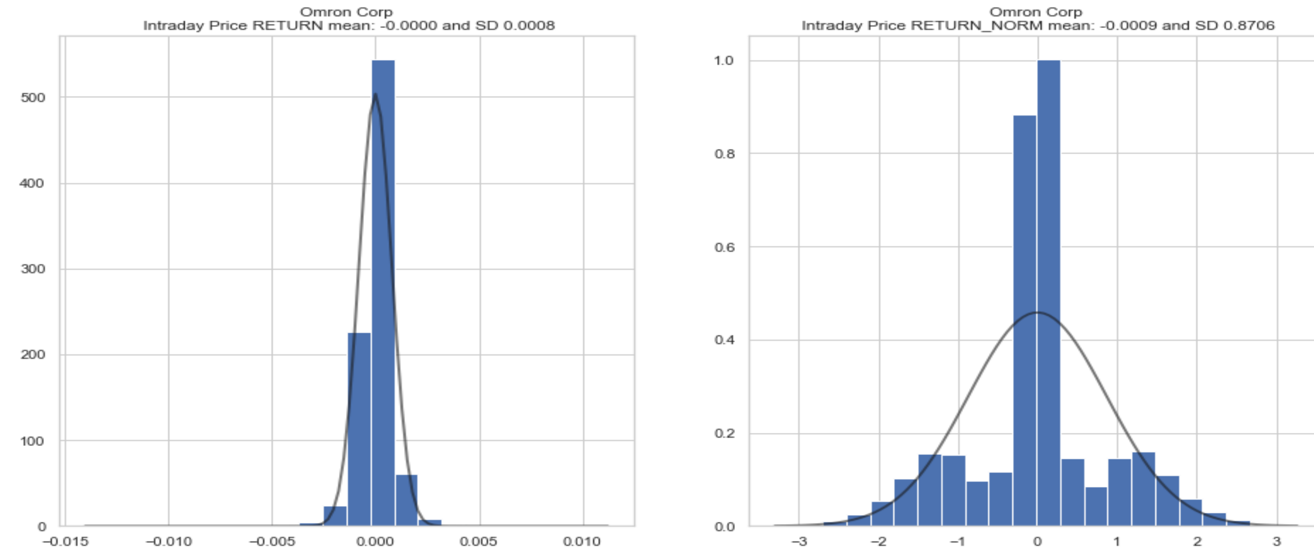
Data comes from the Japan Exchange Group via LSEG's Refinitiv Workspace API. When no trade has taken place during the time interval for individual stock prices, no data is recorded. Ideally, we would record the previous price with 0 volume in this instance, and we have augmented the data in that fashion. Topix 33 Industry sector prices have a small amount of missing data due to price move circuit breakers of sector members or technical issues, and in those cases, we linearly interpolate the missing prices.

We do not include the open and closing prices for the day as they are not intraday prices; the factors influencing them are different and more varied than intraday prices. The time between the close and open can be up to several days vs. the one-minute interval during the day, and during this time, information from financial markets and other sources can cause significant price changes. Similarly, at the close, the need to close out trades or finish customer orders causes volatility of a different magnitude to intraday, as if a different mechanism generated it. We leave the daily morning session close and afternoon open prices as the morning close shows no volatility, and the more volatile afternoon open is still within the range seen in the intraday one-minute intervals. There is nowhere near the level of new information over the lunch break compared to overnight when the European and U.S. markets open.

Large Cap Return Distribution



Small (for Nikkei 225) Cap Return Distribution



Intraday returns for the large-cap, more liquid stocks look close to normally distributed. This distribution is not a surprise as most of the volatility occurs at the open and close, which we decided not to include. However, when we examine the distribution of the intraday returns more closely, we find that they still have longer tails. For less liquid stocks, even intraday, their return distribution strongly reflects traditional stylized facts for asset returns, such as more being more peaked and having heavier tails than the normal distribution.

Data [*normalization*]() is an essential preprocessing step that transforms data attributes with different ranges into a known uniform scale to reduce the complexity inherent in data integrated from multiple sources and contexts. This preprocessing makes comparing different attributes easier and prevents attributes with larger values from dominating over attributes with smaller values. The data becomes better conditioned for convergence leading to better performance by models not explicitly designed to handle inputs with different ranges, and this is necessary as we are comparing across many different methods. We use returns instead of prices for this normalization benefit.

Normalizing data for use in streaming scenarios is challenging because of evolving trends and the non-availability of whole data beforehand. However, we are not using streaming and can use the relatively simple Min-Max type of normalization for scaling.

Returns

The return from the stock in percentage terms is simply the difference in value between the two periods divided by the beginning value.

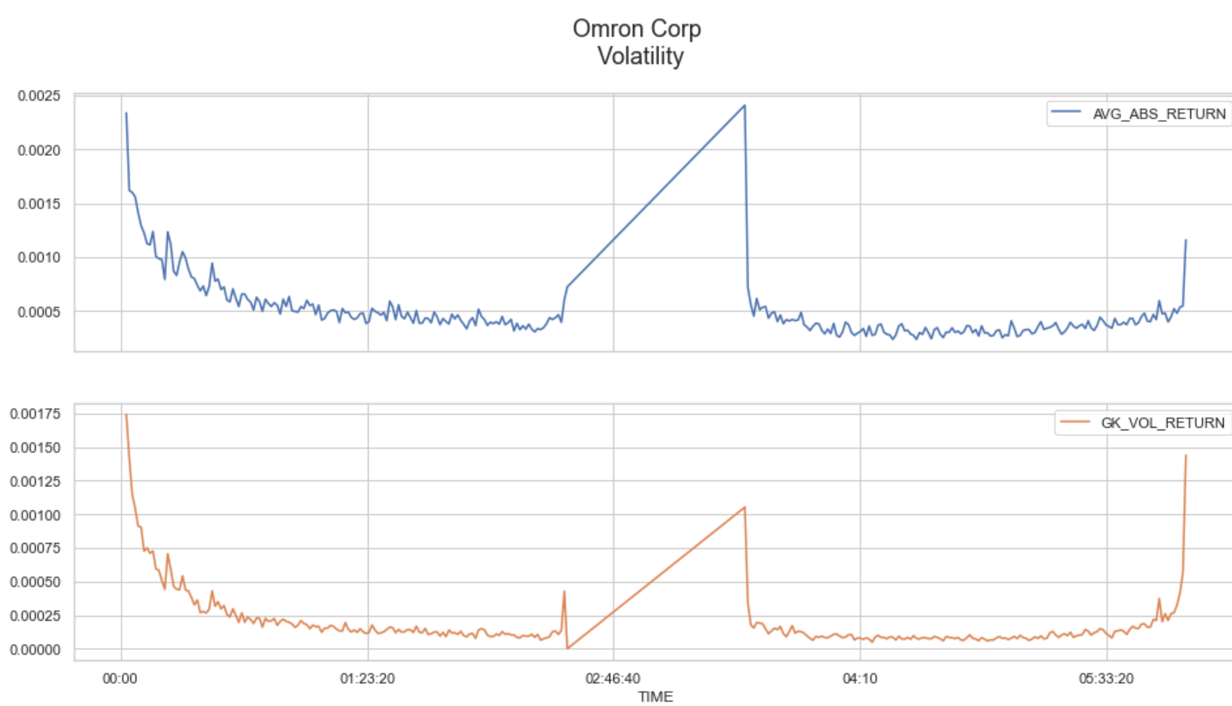
$$R_t = \frac{C_t - C_{t-1}}{C_t}$$

where R_t = Return, C_t = Closing Price

Intraday Volatility Adjustment

Measures

When looking at the intraday volatility chart, we see greater volatility at the open and just before the close. This U-shaped intraday volatility is well known[8] and studied for general securities and stock markets.[9] Additionally, we see volatility at the start of the afternoon session due to the lunchtime close of trading on the TSE, something most equity exchanges do not do. So, the intraday volatility pattern for stocks on the TSE is more of a W shape.



Mean of absolute return

$$\frac{1}{T}\sum_{t=1}^{T}\left|\frac{C_t}{C_{t-1}}-1\right|$$

Garman and Klass (1980)

$$V_{ohlc}=0.5[log(H)-log(L)]^2-[2log(2)-1)][log(C)-log(O)]^2$$

Where
V_{ohlc} is Volatility,
T is time,
C is Closing Price,
H is High Price,
L is Low Price,
O is Opening Price

The average of the absolute return is an easy to understand measure, but because we have the open, high, low, and close data, we use the Garman-Klass estimator. The GK estimator is a commonly used volatility measure and is more effective than the basic formula since it considers the price at the beginning and end of the time interval and intra-minute price extremums. We use $\sigma_{ohlc} = \sqrt{V_{ohlc}}$ to be compatible with convention.
One potential drawback of the GK method is bias from opening jumps in price and trend movements, but we can ignore this drawback as we do not include the opening day price.

Adjusting returns for time of intraday volatility

We assume the trend is null and just divide each return by an empirical estimate of σ,

$$AdjR_t=\frac{R_t}{\sigma_{ohlc_t}}$$

Normalization

Normalization is scaling in which values are shifted and end up in a range between 0 and 1. We use Min-Max scaling for normalization.

$$N = \frac{AdjR - AdjR_{min}}{AdjR_{max} - AdjR_{min}}$$

Post Processed Data

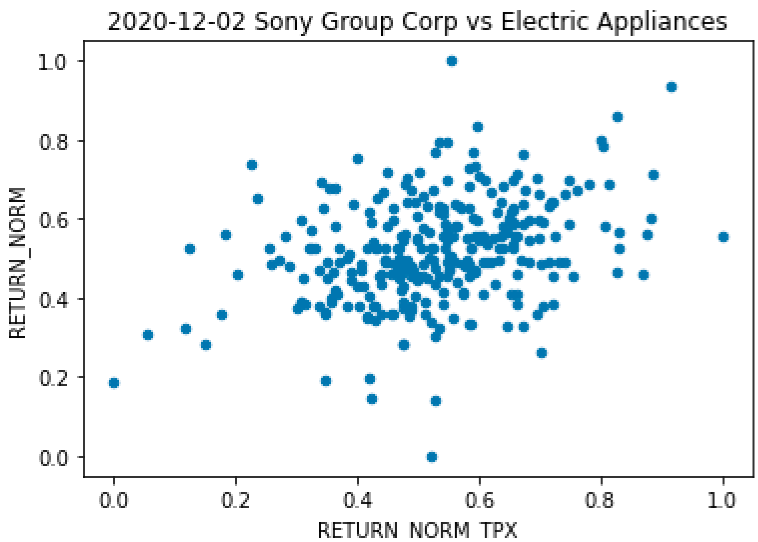
Example of post processed data for Sony.

Date	HIGH	LOW	OPEN	CLOSE	RETURN	ADJ	NORM
00:06:00	9465.0	9430.0	9449.0	9440.0	-0.000635	-0.377085	0.420024
00:07:00	9441.0	9402.0	9441.0	9402.0	-0.004025	-2.671680	0.343129
00:08:00	9417.0	9400.0	9404.0	9409.0	0.000745	0.530153	0.450426
00:09:00	9434.0	9408.0	9410.0	9429.0	0.002126	1.578632	0.485562
00:10:00	9442.0	9423.0	9427.0	9436.0	0.000742	0.584672	0.452253

ARIMA for Close Price is (0,1,0). After adjusting for intraday volatility and normalization, ARIMA is (0,0,0) or close to white noise.

TOPIX Sector Indices

The 33 industrial sectors defined by the Securities Identification Code Committee (SICC) form the TOPIX Sector Indices.[10] As the TOPIX is a superset of the Nikkei 225, all 225 stocks are a constituent of one of the TOPIX sector indices. The stocks correlate with their corresponding Sector Index from an overall market standpoint, and because these sectors are industry sectors, the correlation should be more robust than with the overall market index. This correlation should help signal a contextual anomaly. For example, Honda's stock price falling a moderate amount would not be considered an anomaly, but it would be if the stock price were falling at the same time as the TOPIX Automobile Sector Index was going sharply higher.



Similar to the stock price returns above, we normalize the Topix Sector Indices data, including adjusting returns for intraday volatility.

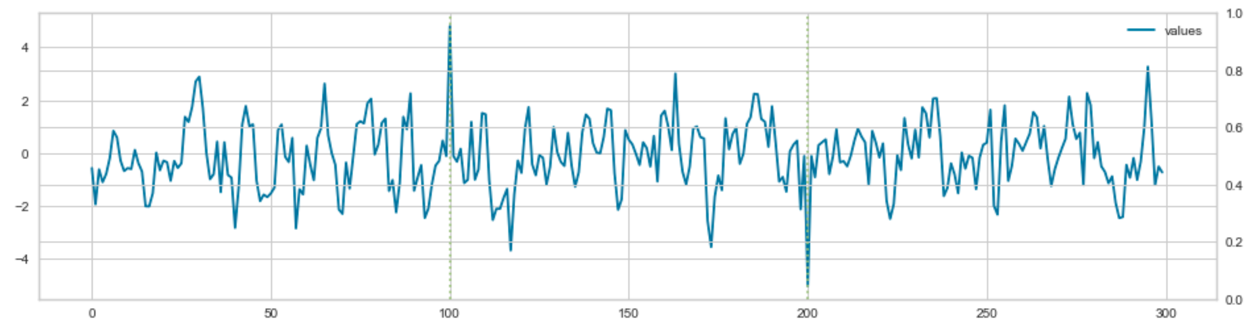
Methodology

Given that the price data set is unlabelled, and considering the impossibility of getting labeled data for actual stock price data that would be widely accepted as correct, we use synthetic data. To keep the resulting data as realistic as possible, we use real stock price data and only inject synthetic outliers while considering possible limitations.[11]

Synthetic Outlier Description

Synthetic Global Outlier

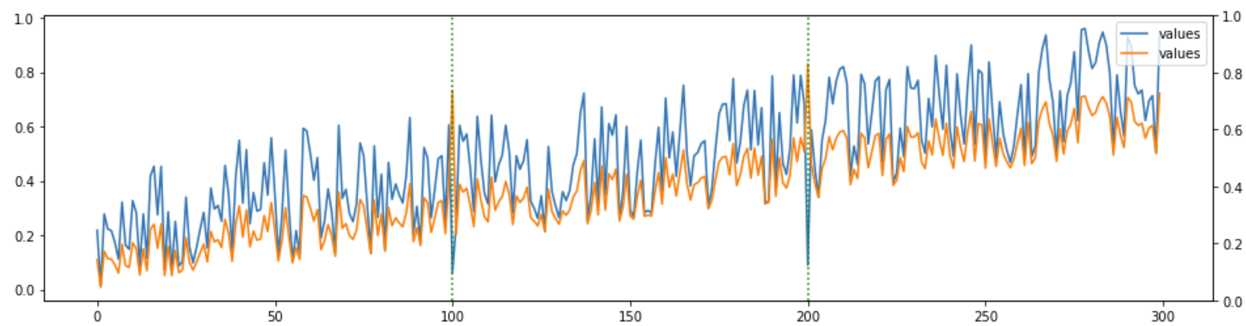
We place the **synthetic global outlier four standard deviations away from the mean**. Four SDs may seem large, but the returns have long tails, and even at 4 SDs, there will be one or two more extreme points than the synthetic global outlier on most days. This synthetic global outlier will be used in a univariate test.



An artificial series with injected synthetic global outliers (points moved four standard deviations away from the mean) at 100 and 200 on the x-axis.

Synthetic Conditional Outlier

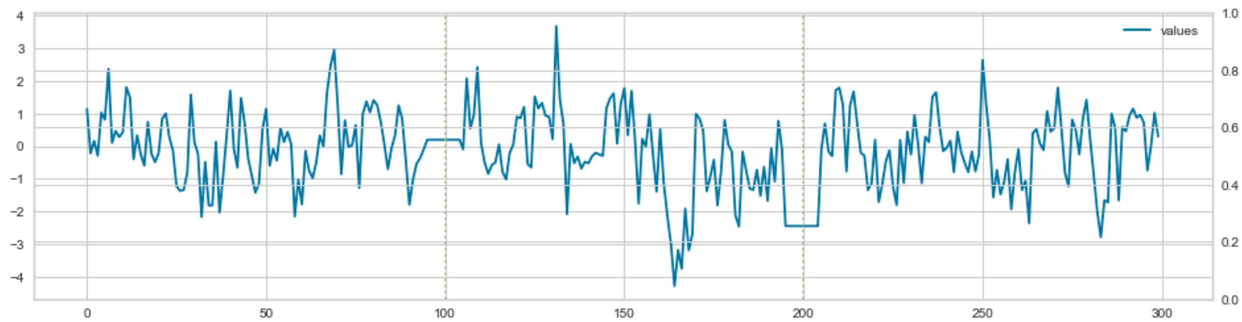
This example of a **synthetic conditional outlier** takes the first and second data points and **moves each one standard deviation away in opposite directions**. This synthetic conditional outlier will be used in multivariate testing.



Artificial example of two correlated series and synthetic conditional outliers injected (points moved one standard deviation away from each other) at 100 and 200 on the x-axis.

Synthetic Collective Outlier

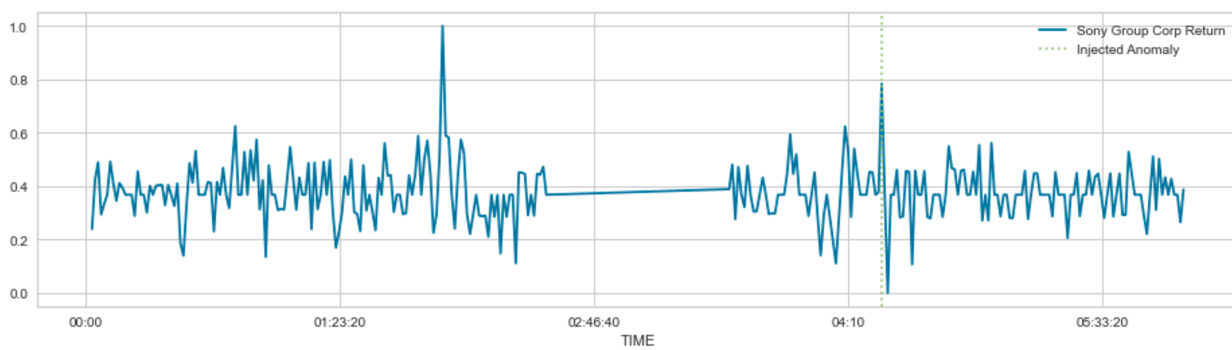
The **synthetic collective outlier is the price from the real data but repeated five times**. An unchanging price is the classic example of a collective outlier for market price data of a liquid security.



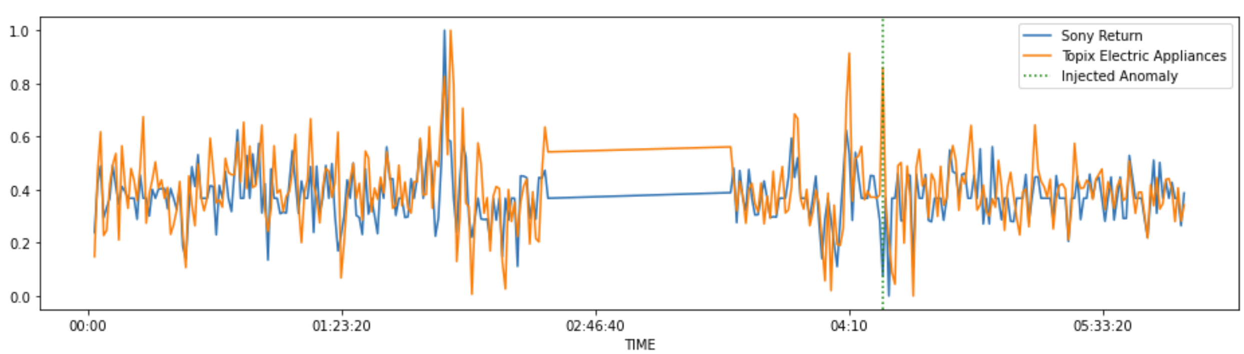
Artificial example of a series with synthetic collective outliers injected (five unchanged points) at 100 and 200 on the x-axis.

Synthetic Outlier in Actual Data

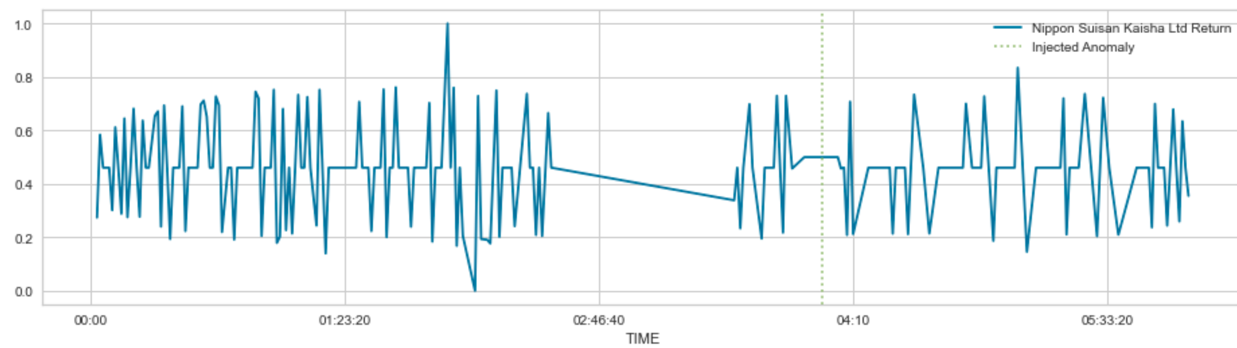
We take two years of daily intraday price data for six stocks (each stock 245 days x 300 data points a day) and inject one of each type anomaly for each day and separately see if we can detect each anomaly.



Example of Global anomaly injected into normalized intraday returns of Sony for 2021-11-16



Example of Contextual Anomaly injected into normalized intraday returns of Sony for 2021-11-16



Example of Collective Anomaly injected into normalized intraday returns of Nippon Suisan Kaisha for 2021-11-16

Given the random nature of stock prices, finding collective outliers is particularly challenging. The difficulty is not just from the point of view of finding them but from the point of view of defining what is a collective outlier as it is both highly contextually dependent and subjective. We use historical volatility as a feature for collective anomalies.[12]

Models

We use standard models as implemented in the Scikit, PMDArima PyOD[13], PySad and PyCaret libraries with as little tuning of hyperparameters as practical. Examples of learned function charts for each model are available in the appendix.

Statistical

(S)ARIMA

First developed almost 80 years ago and commonly applied to financial markets from the 1970s, ARIMA is a forecasting algorithm that uses the differences between its predictions with confidence intervals and actual results for anomaly detection. (S)ARIMA consists of the following terms:

• **AR Auto-regressive** is the expression for the (p) number of lags of the stationarized series

• **I Integrated** is the term used for the (d) number differences needed to make the series stationary.
• **MA Moving average** terms denoting the (q) lags of the forecast errors
• **S Seasonal** portion deals with cycles in the data that repeat at fixed periods and, as a result, requires one additional term (P,D,Q). for each of the three previous terms.

The complete model is the Seasonal ARIMA(p,d,q) x (P,D,Q) model.

Histogram-based Outlier Detection

For each feature, the HBOS uses bins of equal width over the range of the data. Bins with more points are considered normal data, while bins with few data points are considered outliers, and the actual score is calculated as the inverse of the height of the bin. Known for being fast, HBOS works in linear time O(n) in case of fixed bin width or O(n · log(n)) for the variant using dynamic bin widths. Easy to compute for global outliers, this method has trouble finding local outliers and assumes feature independence.

Proximity, Clustering and Angle Based Methods

k-NN Distance

This method measures the distance to its *k*-nearest neighbors. The number *k* must be manually specified, and the chosen number can make a considerable difference. In this paper, we use the hyperparameters provided by the PyCaret library with n_neighbours = 5 and a radius of 1.0. The output number will vary considerably based on the normalization of the data and dimensions, so ranking is necessary.

Local Outlier Factor

The original paper for LOF was published in 2000, and although it is not the latest technology, it is still an attractive method because of its practical detection performance and simple implementation.

LOF focuses on the density of data in space. In particular, it focuses on local density, which measures how dense a point is with its *k* nearest neighbors. The *k*-neighborhood refers to the *k* points that are closest to a point. This measure allows LOF to detect anomalies that histograms would miss. Global anomalies can still be detected since they also have a low local reachability density (LRD). In theory, using LOF to detect global outliers could be problematic because of all the false positive local outliers that are detected and this has reportedly been the case for some uses in practice. However, this was not an issue for detecting the synthetic outliers we introduced into Japanese stock market prices.

Clustering-Based Local Outlier Factor

CBLOF is another anomaly detection algorithm derived from KNN and LOF but uses clustering for determining outliers. First clusters are found (often using *k*-means for performance reasons), then calculating the density for every cluster.

Data may be clustered in several different places to form clusters. When a point is closer to a large cluster, the higher the probability that this is a normal point, and vice versa, the farther away a point is from a large cluster, the lower the probability.

As with the previous nearest-neighbor-based algorithms, choosing the number k of initial clusters *k* can significantly affect the results.

Angle-based Outlier Detection

Uses angles for measuring distances which provides better performance at higher dimensions.

Classification based methods

Isolation Forest

The data is partitioned into normal and outliers by measuring how isolated a leaf is in the forest. I-Forest is another algorithm that is supposed to work well with higher dimensions.

One-class SVM detector

Initially proposed in 2001 for semi-supervised global anomaly detection, OC-SVM methods have since been applied to unsupervised detection. One-class SVMs separate the origin from the kernel space data instances, resulting in complex hulls describing the usual data in the feature space. Subsequently, a normalized distance to the determined decision boundary scores every data point, and a soft margin is used to decide which points are anomalies. One-class SVM running time grows $O(n^2)$.

Dimensionality Reduction

Principal Component Analysis

PCA reduces data dimensionality by compressing the feature space into a smaller subspace that retains most of the information in the original. The principal components found are the eigenvectors of the covariance matrix, so PCA encodes linear relationships. PCA assumes that the distance between different series uses euclidean distances.

Neural Networks

Autoencoder

Autoencoder methods compress the feature space into a smaller subspace using neural networks and can be included in both the Dimensionality Reduction and Neural Networks categories. As an autoencoder uses a neural network to reduce dimensionality, it can capture both linear and nonlinear transformations and, in theory, should be capable of better results than PCA.

Deep SVDD

The Deep Support Vector Data Description (Deep SVDD)[14] uses a neural network to minimize a hypersphere that contains the data. The minimization function causes the model to extract common factors to place the data points close to the origin of the hypersphere. Those points that the neural network cannot position near the center are outliers.

Metrics for Assessment

Many of the methods are non-deterministic, particularly the KNN related methods, but we offset that drawback by running them approximately 250 times for each of the six stocks for each of the anomaly types.

The confidence numbers output by the various detection methods for outliers are not directly comparable to each other, so the relative rankings of each of the points are used instead.

The synthetic datasets have labeled outliers, so it is easy to use standard metrics such as Accuracy, Precision, Recall, and the F-measure. For collective anomalies, we use a slightly offset window of the same length as the anomaly for being correct.

Results

Average across 8 stocks, daily over 2 years

Global Anomaly

Model	Accuracy	Precision	Recall	F1 Score
knn	0.999039	0.815279	1.0	0.897693

iforest	0.998963	0.813837	1.0	0.896602
svm	0.998450	0.777602	1.0	0.868178
arima	0.998623	0.750446	1.0	0.855527
abod	0.997822	0.689876	1.0	0.812515
lof	0.994849	0.675201	1.0	0.775532
cluster	0.997048	0.602122	1.0	0.750924
pca	0.996610	0.555060	1.0	0.708534
DeepSVDD	0.994776	0.471587	1.0	0.640085
AE	0.994786	0.458680	1.0	0.628271
histogram	0.993949	0.423370	1.0	0.594409

Contextual Anomaly

Model	Accuracy	Precision	Recall	F1 Score
iforest	0.998724	0.799572	1.0	0.886549
knn	0.998337	0.788088	1.0	0.874472
abod	0.998238	0.783913	1.0	0.870551
lof	0.997833	0.777199	1.0	0.860879
arima	0.995073	0.670831	1.0	0.769501
cluster	0.996516	0.629327	1.0	0.763246
svm	0.996103	0.601756	1.0	0.741278
pca	0.992761	0.387573	1.0	0.556133
histogram	0.992755	0.388018	1.0	0.555456
AE	0.990815	0.354416	1.0	0.517990
DeepSVDD	0.990003	0.335864	1.0	0.499737

Collective Anomaly

Model	Accuracy	Precision	Recall	F1 Score
arima	0.994680	0.718870	1.0	0.793173
lof	0.992694	0.650509	1.0	0.746440
cluster	0.987904	0.564828	1.0	0.673520
iforest	0.992275	0.475268	1.0	0.626052
AE	0.976084	0.528298	1.0	0.623571
pca	0.977878	0.502902	1.0	0.609872
DeepSVDD	0.968475	0.495256	1.0	0.586734
knn	0.983098	0.435975	1.0	0.567084
svm	0.954643	0.438156	1.0	0.536127
histogram	0.983636	0.227538	1.0	0.368003
abod	0.975308	0.180306	1.0	0.301999

The collective anomaly detection of the highest performing models is sensitive to the historical volatility hyperparameter.

Combined Anomalies

Results detecting all three different types of outliers using a single model for Sony over two years.

Model	Stock	Accuracy	Precision	Recall	F1 Score
arima	6758.T	0.995283	0.829932	0.666667	0.739394
lof	6758.T	0.994241	0.734940	0.666667	0.699140
knn	6758.T	0.993925	0.710335	0.666667	0.687808
abod	6758.T	0.992691	0.628057	0.666667	0.646786
knn	6758.T	0.988207	0.459799	1.000000	0.629948
iforest	6758.T	0.992115	0.595849	0.666667	0.629271

lof	6758.T	0.987782	0.451017	1.000000	0.621656
lof	6758.T	0.987782	0.451017	1.000000	0.621656
pca	6758.T	0.991772	0.578199	0.666667	0.619289
AE	6758.T	0.991649	0.572098	0.666667	0.615773
DeepSVDD	6758.T	0.991635	0.571429	0.666667	0.615385
cluster	6758.T	0.991430	0.561565	0.666667	0.609619
iforest	6758.T	0.986479	0.426077	1.000000	0.597551
iforest	6758.T	0.986479	0.426077	1.000000	0.597551
pca	6758.T	0.983641	0.380260	1.000000	0.550997
pca	6758.T	0.983641	0.380260	1.000000	0.550997
cluster	6758.T	0.982818	0.368766	1.000000	0.538830
svm	6758.T	0.988139	0.440036	0.666667	0.530147
AE	6758.T	0.979692	0.330773	1.000000	0.497114
arima	6758.T	0.993226	0.976000	0.333333	0.496945
abod	6758.T	0.979596	0.329730	1.000000	0.495935
lof	6758.T	0.993171	0.960630	0.333333	0.494929
DeepSVDD	6758.T	0.979500	0.328693	1.000000	0.494762
knn	6758.T	0.993048	0.927757	0.333333	0.490452
cluster	6758.T	0.992856	0.880866	0.333333	0.483647
svm	6758.T	0.992760	0.859155	0.333333	0.480315
abod	6758.T	0.992664	0.838488	0.333333	0.477028
pca	6758.T	0.992458	0.797386	0.333333	0.470135
iforest	6758.T	0.991745	0.681564	0.333333	0.447706
histogram	6758.T	0.974687	0.233493	0.666667	0.345854
histogram	6758.T	0.986946	0.344633	0.333333	0.338889
histogram	6758.T	0.949154	0.164865	1.000000	0.283063
svm	6758.T	0.948551	0.163247	1.000000	0.280675
svm	6758.T	0.948551	0.163247	1.000000	0.280675
arima	6758.T	0.850424	0.062887	1.000000	0.118332

Conclusion

We get satisfactory results for anomaly detection of intraday stock prices using standard models with minimal manual tuning and a modest amount of domain knowledge to set up the data. The results are acceptable considering the unsupervised nature of the task, that stock returns are not necessarily IID, and the anomalies are complicated for even a human to pick out by looking at a graph. Four observations stand out:

Better results and easier explainability are obtained by explicitly setting up a model for each of the three types of global, conditional, and collective outliers compared to trying to use the same model to detect all three.

There are many methods for anomaly detection with different strengths and weaknesses, and there is significant variability in the results depending on the individual stock. Consequently, among the methods for detecting a particular type of outlier, it is hard to declare which particular method among the top four is the clear winner.

In general, anomaly detection works better for larger-cap, more liquid stocks, and the difference is significant. Large-cap stocks performed 0.2 better on their F1 score than the smaller-cap equities that we tested. See appendix B. This performance gap needs further examination as what is considered a small-cap for the Nikkei 225 is a mid-cap for the broader Japanese market (Topix).

In other areas of machine learning, such as vision and NLP, neural networks have started to dominate the top results. This superior performance by NNs is not the case with detecting anomalies in intraday stock prices. Our results showing the particular NN models in the library we tested not to be competitive align with results from papers in other areas of anomaly detection.[15] We do not have the evidence to go as far as to say statistical methods are clearly better than machine learning methods in general[16], as some others have suggested.

The author is unaware of any results for intraday anomaly detection to compare with the above results.

Future areas for study include the effect of different types of normalization on results and why detecting outliers is significantly affected by the stock's liquidity.

List of used sources

Bibliography

1. D. Hawkins. Identification of Outliers, Chapman and Hall, 1980.

2. Krawczyk, B. "Learning from imbalanced data: open challenges and future directions." *Prog Artif Intell*, vol. 5, pp. 221–232, 2016, https://doi.org/10.1007/s13748-016-0094-0.

3. Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems)* (3rd ed.). Morgan Kaufmann. P. 545, 2012

4. Mitsa, Theophano. Temporal data mining. CRC Press, 2010

5. Thakkar, P., Vala, J., & Prajapati, V., *Survey on Outlier Detection in Data Stream*, 2016

6. Pang, G., Shen, C., Cao, L., & Hengel, A. V., *Deep Learning for Anomaly Detection: A Review*, 2020

7. Kriegel, Hans-Peter, et al. "Angle-Based Outlier Detection in High-dimensional Data." *KDD*, 2008, https://www.dbs.ifi.lmu.de/Publikationen/Papers/KDD2008.pdf.

8. Gabrys, R., Hörmann, S., Piotr, K., *Monitoring the Intraday Volatility Pattern*. Journal of Time Series Econometrics, January 2013

9. Li, Young, Bloomberg Enterprise Quants, *A Practical Model for Prediction of Intraday Volatility*, 2021

10. Japan Exchange Group, Inc. (JPX). "東証指数算出要領（東証業種別株価指数・TOPIX-17 シリーズ編）." *JPX*, 13 August 2013, https://www.jpx.co.jp/markets/indices/topix/tvdivq00000030ne-att/9_sector.pdf.

11. Steinbuss, Georg, and Klemens BÖHM. "Benchmarking Unsupervised Outlier Detection with Realistic Synthetic Data." *arxiv*, 2020, https://arxiv.org/pdf/2004.06947.pdf.

12. Lai, Kwei-Herng, et al. "Revisiting Time Series Outlier Detection: Definitions and Benchmarks." 2021, https://openreview.net/pdf?id=r8IvOsnHchr.

13. Zhao, Y., Nasrullah, Z. and Li, Z.,. PyOD: A Python Toolbox for Scalable Outlier Detection. Journal of machine learning research (JMLR), 20(96), 2019 pp.1-7.

14. Ruff, Lukas, et al. “Deep One-Class Classification.” *Proceedings of Machine Learning Research*, vol. 80:4393-4402, 2018, https://proceedings.mlr.press/v80/ruff18a.html.

15. Braei, Mohammad and Wagner, Sebastian. Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art, Research Gate, p. 22, 2020, https://www.researchgate.net/publication/340374826_Anomaly_Detection_in_Univariate_Time-series_A_Survey_on_the_State-of-the-Art/fulltext/5e855ad092851c2f52744095/Anomaly-Detection-in-Univariate-Time-series-A-Survey-on-the-State-of-the-Art.pdf

16. Makridakis, P., Spiliotis, E.,, Assimakopoulos, V., *Statistical and Machine Learning forecasting methods: Concerns and ways forward*, 2018. P. 20 (STATs vs New Methods)

Libraries

Numpy
Pandas
Scikit
PMDArima
PyOD A Python Toolbox for Scalable Outlier Detection
PySad Streaming Anomaly Detection Framework in Python
PyCaret Low touch Python machine learning library
River A Python package for online/streaming machine learning.
Unsupervised clustering.

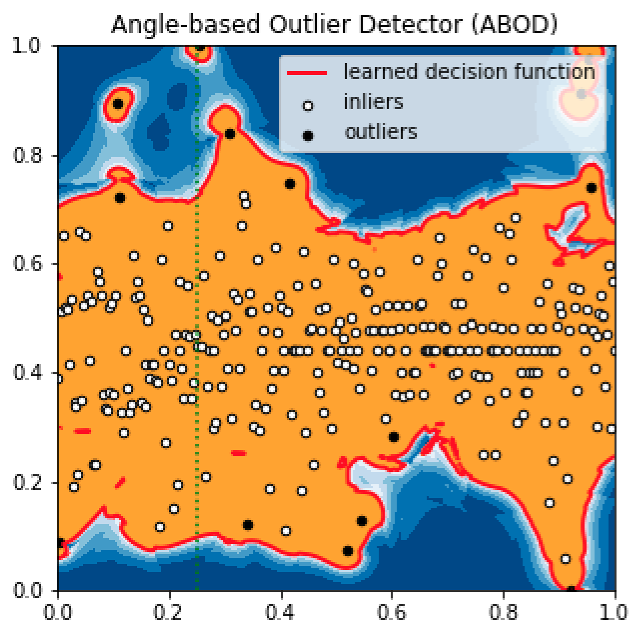
Applications

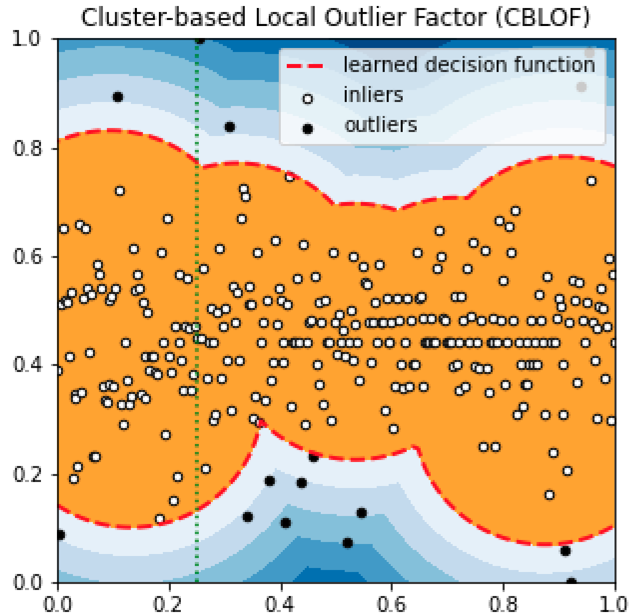
https://github.com/JamesSullivan/ad_test

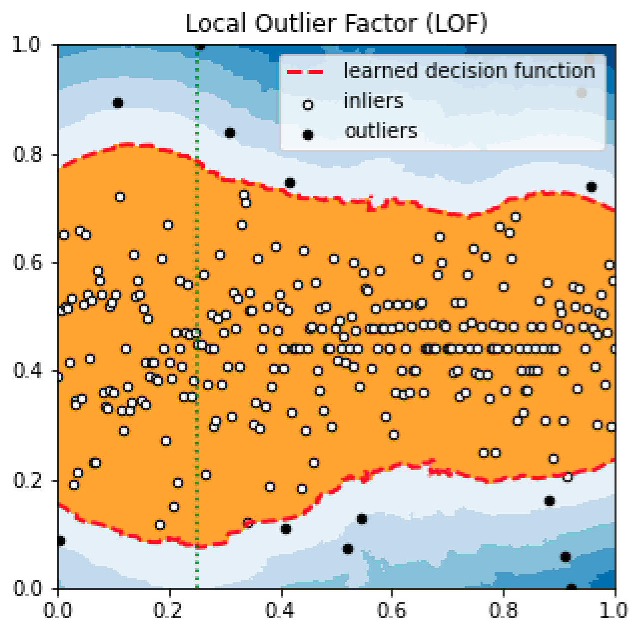
Appendices

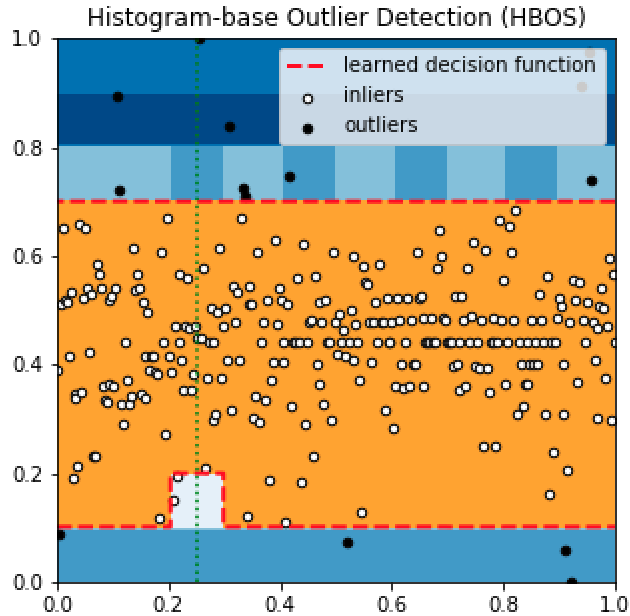
A. Learned decision function charts

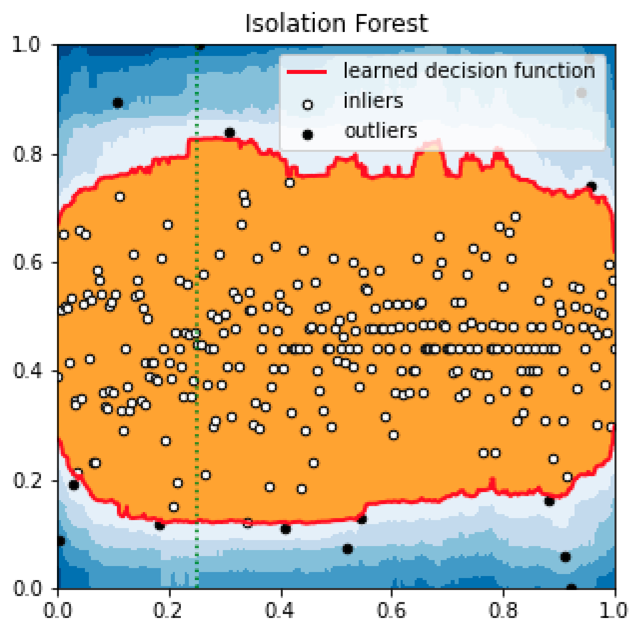
Various anomaly detection methods for Sony with global outlier injected at 0.25 (green dotted line to outlier at top of chart) using PyOD library with the outlier factor set to 0.05.

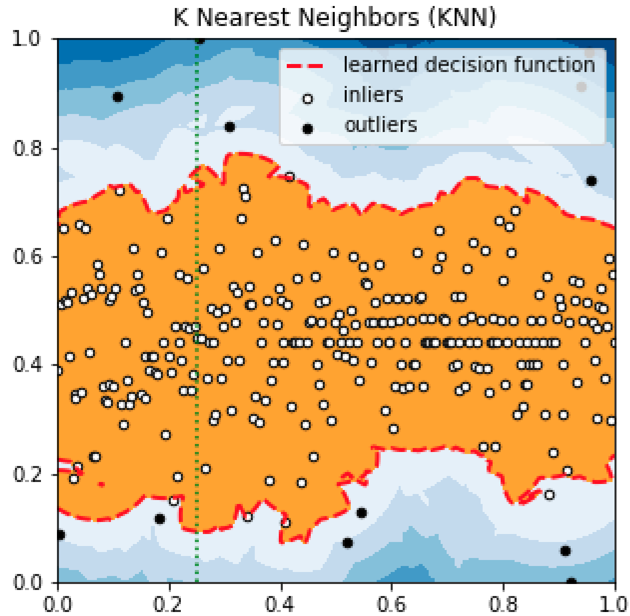



Cluster-based Local Outlier Factor (CBLOF)
learned decision function
inliers
outliers
0.0
0.2
0.4
0.6
0.8
1.0

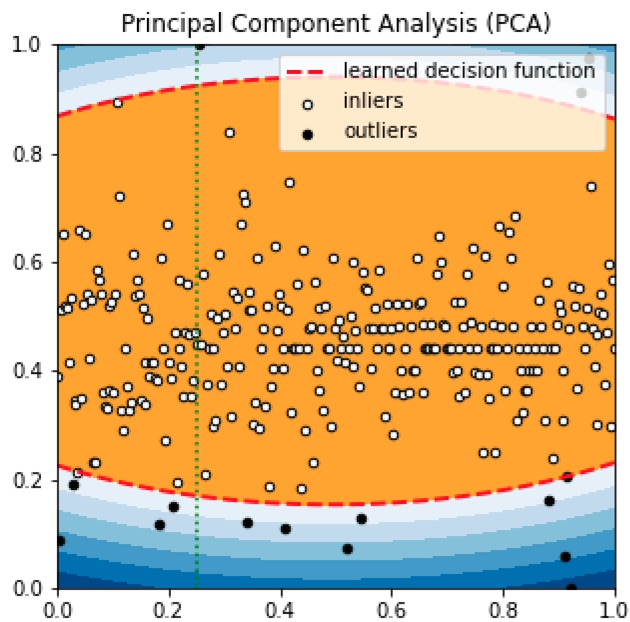

Local Outlier Factor (LOF)
learned decision function
inliers
outliers
0.0
0.2
0.4
0.6
0.8
1.0


Histogram-base Outlier Detection (HBOS)
learned decision function
inliers
outliers
0.0
0.2
0.4
0.6
0.8
1.0

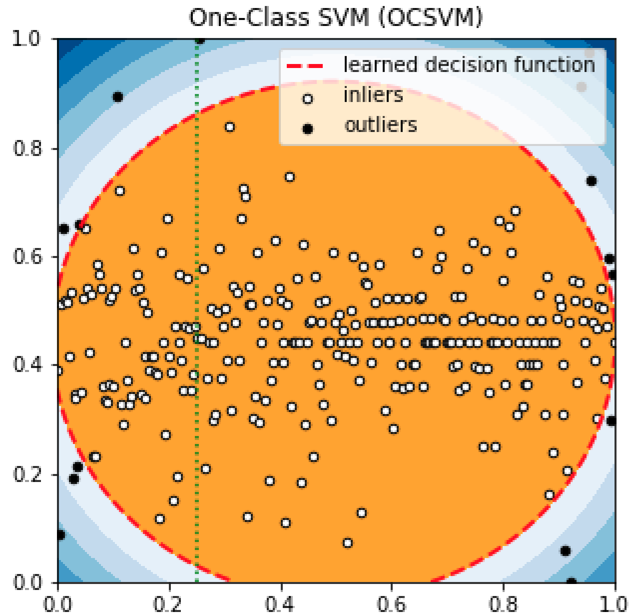

Isolation Forest
learned decision function
inliers
outliers
0.0
0.2
0.4
0.6
0.8
1.0

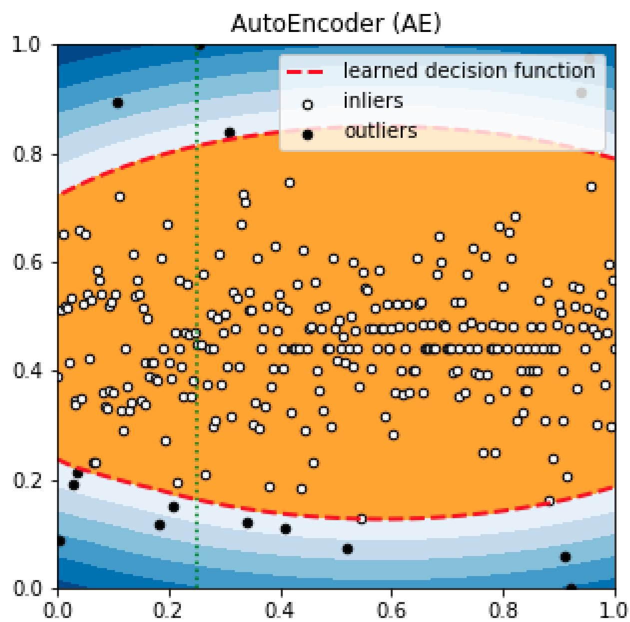


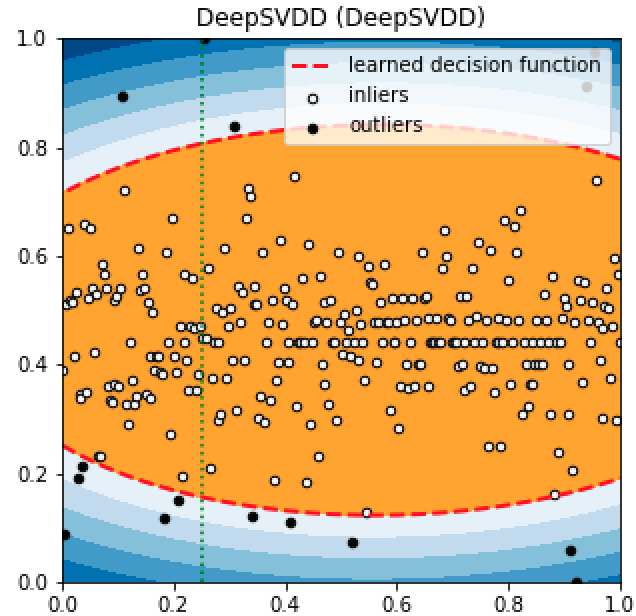
K Nearest Neighbors (KNN)
learned decision function
inliers
outliers
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0



Principal Component Analysis (PCA)
learned decision function
inliers
outliers
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


One-Class SVM (OCSVM)
learned decision function
inliers
outliers
0.0
0.2
0.4
0.6
0.8
1.0


AutoEncoder (AE)
learned decision function
inliers
outliers
0.0
0.2
0.4
0.6
0.8
1.0



B. Further Results

Large Capitalization Stocks

Average of four largest capitalized stocks included in Nikkei 225: '6098.T', '6758.T', '7203.T', '9432.T'

Model	Accuracy	Precision	Recall	F1 Score
lof	0.999516	0.875904	1.0	0.933266
arima	0.999247	0.839791	1.0	0.906741
iforest	0.998832	0.763545	1.0	0.860340
svm	0.998811	0.761543	1.0	0.858529
knn	0.998808	0.759468	1.0	0.857663
cluster	0.998418	0.693085	1.0	0.814737

abod	0.995487	0.619791	1.0	0.718958
AE	0.996744	0.559305	1.0	0.700021
DeepSVDD	0.996671	0.557917	1.0	0.697440
pca	0.996580	0.530861	1.0	0.681319
histogram	0.993006	0.358292	1.0	0.518520

Small Capitalization Stocks

Average of four smaller capitalized stocks included in Nikkei 225: '1332.T', '3103.T', '5707.T', '6703.T'

Model	Accuracy	Precision	Recall	F1 Score
iforest	0.994475	0.628907	1.0	0.745796
arima	0.993004	0.586974	1.0	0.705393
knn	0.988175	0.600094	1.0	0.701836
lof	0.990735	0.526035	1.0	0.655301
cluster	0.989228	0.504433	1.0	0.643723
abod	0.985426	0.482939	1.0	0.604418
svm	0.967319	0.450133	1.0	0.571859
pca	0.981586	0.432828	1.0	0.568374
histogram	0.987220	0.334325	1.0	0.493392
DeepSVDD	0.972879	0.336253	1.0	0.474852
AE	0.975991	0.332374	1.0	0.473497