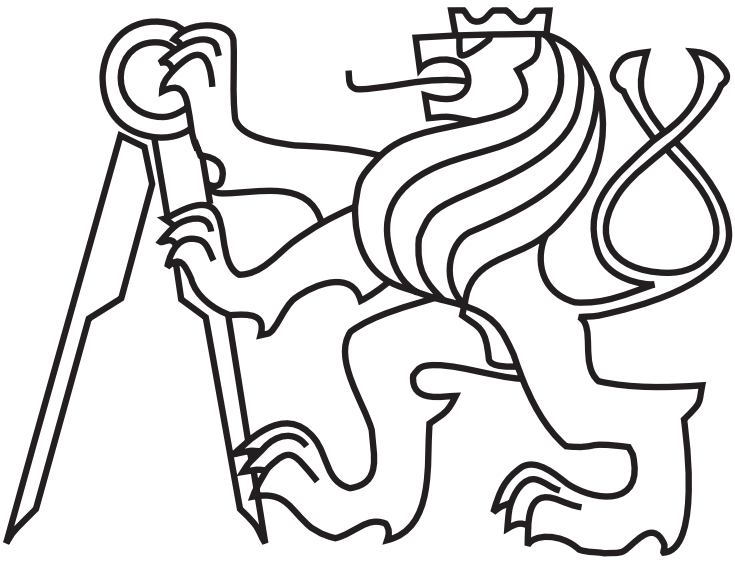
CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

BACHELOR THESIS



Martin Rektoris

Anomaly Detection in Periodic Stochastic Phenomena

Department of Control Engineering

Thesis supervisor: **Ing. Tomáš Vintr**

May, 2021

Prohlášení

I hereby declare that I have completed this thesis independently and that I have used only the sources (literature, software, etc.) listed in the enclosed bibliography.

In Prague on..... .....

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Rektoris Martin** Personal ID number: **483559**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Control Engineering**

Study program: **Cybernetics and Robotics**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Anomaly detection in periodical stochastic phenomena

Bachelor's thesis title in Czech:

Detekce anomálií v periodických stochastických jevech

Guidelines:

1) Research forecasting methods used in the mobile robotics domain.
2) Research methods for outlier and anomaly detection.
3) Select or design suitable methods and criteria to asess the performance of anomaly detection methods.
4) Select a set of scenarios and datasets to apply the chosen methods.
5) Design and create reproducible experiments.
6) Evaluate outlier detection methods in selected scenarios.

Bibliography / sources:

[1] KRAJNÍK, Tomáš, et al. Warped hypertime representations for long-term autonomy of mobile robots. IEEE Robotics and Automation Letters, 2019, 4.4:3310-3317.
[2] BREUNIG, Markus M., et al. LOF: identifying density-based local outliers. In: Proceeding of the 2000 ACM SIGMOD international conference on Management of data. 2000.p.93-104.

Name and workplace of bachelor's thesis supervisor:

Ing. Tomáš Vintr, Artificial Intelligence Center, FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

doc. Ing. Tomáš Krajník, Ph.D., Artificial Intelligence Center, FEE

Date of bachelor's thesis assignment: **15.01.2021** Deadline for bachelor thesis submission: **21.05.2021**

Assignment valid until:
by the end of summer semester 2021/2022

Ing. Tomáš Vintr	prof. Ing. Michael Šebek, DrSc.	prof. Mgr. Petr Páta, Ph.D.
Supervisor's signature	Head of department's signature	Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student's signature

Acknowledgements

I would first like to thank my supervisor Tomáš Vintr for offering me an opportunity to work on this exciting topic and for many hours of fruitful discussion. Thanks go to head Chronorobotics laboratory Tomáš Krajník who introduced me to Chronorobotics laboratory two years ago.

I want to thank my family, Terka and Matouš for all the support and patience with me while working on this thesis.

Abstract

Chronorobotics provides spatio-temporal forecasting tools that were successfully applied in the field of autonomous robotics. However, these methods do not provide an appropriate tool to detect novelty. This thesis concerns suitable tool for novelty and generally outlier detection for this scientific field. It provides research of suitable methods from both fields, combines them and evaluates their combinations in the experiments. Although some of them show good quality on synthetic time-series, their application to real data reveals the necessity of further development.

Abstrakt

Chronorobotika nabízí nástroje pro předpovědi v čase a prostoru, které byly úspěšně použity v autonomní robotice. Nicméně, tyto metody nenabízí vhodné nástroje pro detekci nových jevů. Tato práce se zabývá vhodnými nástroji pro detekci nových jevů a obecně detekci odlehlých pozorování v tomto oboru. V této práci jsou také zkoumány vhodné metody z obou oborů, které jsou kombinovány a jejich kombinace jsou vyhodnoceny v experimentech. Ačkoli některé metody se zdají být velmi slibné na syntetických časových posloupnostech, jejich aplikace na reálných datech ukazuje, že je nezbytný další vývoj.

Contents

List of Figures

List of Tables

1 Introduction

In the thesis, I am concerned with applying anomaly detection methods regarding Chronorobotics principles [1]. The original idea was to detect outliers in spatio-temporal data while modelling time-space together. The research topic was derived from the current issues in the mobile robotics domain, where autonomous robots are expected to deal with the dynamics of the human-populated environment.

Last year, the social setup changed in a way that thwarts the regular human behaviour datasets collection. It was expected to gather the data from the corridors of different universities, but universities were closed. We arrange data collection from the factory, where labours work in shifts, but it was closed due to the spread of disease and never restored to full-fledged running. We were forced to change the data collection, which heavily influenced this thesis. The data we are collecting now lacks the immediate spatiotemporal context. The spatial context can be gathered from the global position and parameters of different places, which is beyond the scope of this bachelor thesis. Therefore, the data analysed in the experiments are pretty common data-series, which do not highlight the main advantages of chronorobotics forecasting methods. On the other hand, the collected data are small, sparse, and irregularly acquired, which prevent the mainstream approaches based on neural networks and developed for big data from being applied.

The lack of the human behaviour datasets that include labelled outliers led me to show the properties of the compared methods on synthetic time-series created according to the only type of data we are gathering now. The real dataset consists of ordered classes describing the relative crowdedness of different places irregularly measured during few weeks. Such data collection can represent an introductory scenario for a service robot in a shopping mall. Let us assume that the service robot has its tasks, but it can also perceive its surroundings. It does not have time to go through all places and count all people at every place, but it can estimate the relative crowdedness in a similar way as it was defined in the FreMEn contra COVID project. After few days of the extensive model building, it can detect suspicious situations, include rare events into its schedule, or improve its recommendation system in a way that will not send customers to unexpectedly crowded places.

In the experiments, different forecasting and outlier detection methods are combined and tested over the vast amount of synthetic time-series generated in chosen fashion. Then I analyse the ability of different combinations of methods to predict the outliers with a particular focus towards finding the suitable value of the threshold, the main parameter that divides outliers and inliers. The methods are then applied to the real datasets with chosen threshold, and the evaluation of the results is provided. It was shown that the complexity of human behaviour with sparsity and irregularity of proposed data collection that simulates random exploration of a service robot led to a general inability of tested methods to provide considerably good outlier detections.

2 Forecasting methods

2.1 Motivation

The advances in autonomous robotics allowed the deployment of robots in a human-populated environment [2]. This environment changes dynamically according to human actions [3], daytime [4], and seasons [5]. For the robot to operate long-term in this environment, it is crucial to incorporate these dynamics into its model [6] which is used for localisation, mapping, and navigation [7]. Therefore, we cannot neglect the dynamics represented by the temporal part [8]. We need to analyse the data features (position, velocity, and others) and timestamps of measurements of features together. If we try to incorporate time in our model, we can encounter some problems due to its nature [9]:

1. The first problem is that time is "infinite", and we cannot measure it to the "end".
2. The second problem is that time is unrepeatable, and we cannot measure the same point twice.

Most state-of-the-art methods cannot deal with these problems, especially on sparse datasets with unevenly spaced measurements.

Krajník et al. [10] addressed the issues mentioned above and came up with the idea of using the frequentist approach for modelling temporal data. The most recently presented method is called Warped Hypertime [11]. It transforms "limitless" time to a bounded multidimensional vector space, which can be analysed using standard statistical methods or more advanced directional statistics methods [12]. The Warped Hypertime method has shown to improve robot localisation, navigation and mapping in the long term.

Additionally, it was shown that some forecasting methods used in chronorobotics [13] could be used other than the robotic domain. For example, in prediction demand [14] or recommendation algorithm domain [15].

2.2 General approach to time-modelling

Timeseries is usually analysed using decomposition into components [16]. Having time-series $f(t)$ for some timestamps t, the series can be decomposed into the trend, seasonality, and noise. These three components are combined depending on the nature of the model. The combination can be additive or multiplicative.

[Regresive methods] Modelling the trend can be performed using standard regression methods such as linear regression [17] or support vector regression [18], which commonly serve as baseline methods. Another commonly used methods are autoregressive forecasting methods [19]. These methods work with stationary time-series or time-series that are stationary after a procedure called "differencing". It consists of methods such as ARIMA,

SARIMA or STARIMA. However, these methods also require to work with time-series as sequences, which are chronologically ordered and have equally sized timestamps. The solution to uneven steps might be an interpolation of missing values when only a few of them. This approach fails with large and frequent gaps in data, which is quite common in many cases. The main problem of approaches to the time series forecasting that expect regular steps is their ability to predict "few steps into the future", which leads to their inability to predict values in specific timestamps. Although sequential forecasting is under heavy developement [20], it was shown to be impossible to apply that on non-sequential timeseries [21].

Apart from the tools to analyse sequences, there is a time-series forecasting tool called Prophet [22]. The forecasting method is based on an additive model with a nonlinear trend with seasonal components represented by the Fourier series. The method is fitted using a probabilistic approach - a maximum aposteriori estimate. Bayesian inference is performed for the normal distribution parameters that are centred around the model's curve. The fitted curve is the mean of the normal distribution given some timestamp. Moreover, every parameter of this method has its prior distribution, enabling Prophet to fully automated model-fitting and forecasting. Contrary to other methods like ARIMA, Prophet is robust to data with unevenly spaced timestamps. This method represents the state-of-the-art for one-dimensional time-series forecasting.

2.3 Chronorobotics approach to time-modelling

Chronorobotics present an approach to spatiotemporal modelling that focuses on modelling space and time together, not as separated entities. This approach has been shown to improve long-term prediction and enable robots to operate in a given environment for a long time [8]. The approach says that in the robotics domain, the trend can be neglected [23] and that it is sufficient to model only periodic characteristicsi [24].

Frequency Map Enhancement The first Chronorobotics model is Frequency Map Enhancement [10], which was presented to model environmental dynamics. This method discretises a spatio-temporal space into a cell, where each cell represents the state of the cell - cell is either occupied or not. Used methods were applied in hospital in Austria to help service robot plan its way [25].

Warped Hypertime Warped Hypertime [26] has been applied to multiple scenarios. The work of Kubis [14] successfully combined methods from the Chronorobotics domain with the prediction demand domain, which resulted in the creation of new spatio-temporal models, as well as in the new area of possible applications. Work [2] of Vintr focuses on prediction direction, speed of pedestrian flows over time and space.

Novel Approaches There also was more theoretical work of Menzl [12] which focuses on the improvement of the currently used Chronorobotics method employing directional statistics. The author presented multiple novel approaches to spatio-temporal modelling. However, the most promising ones use a method of moments as a method for distribution's parameter estimation. It results in a system of nonlinear equations, which does not always have a solution, and therefore is unstable.

2.4 Evaluation of forecast

The usual way to evaluate forecasting quality is a family of measures derived from "mean square error", such as RMSE, MAE, MAPE [16]. Although these measurements are a usual part of toolboxes and manuals, there exists protracted debate about their general applicability [27]. Different authors proposed more suitable measures like Geometric Mean of the Relative Absolute Error and Median Absolute Percentage Error that reflects the relationship of evaluation to decision making [28, 29]. Hyndman et al. [30] proposed Mean Absolute Scaled Error, which on the other hand, can be used only for forecasting sequences.

Chronorobotics faced the simmilar issues [2]. The thesis of Filip Kubis [14] designs appropriate criteria for demand forecasting. The presented evaluation metric, called Random Area, deals with issues that arise while comparing discrete and continuous models. Vintr et al. [24] proposed two evaluation criteria, Total encounters and Expected encounters derived from the "service disturbance" distribution. Similarly to [28] they stated that the measurement has to reflect the purpose of the forecasting. Although the criteria were derived for the specialised task, Expected Encounters (EE) can be applied to different forecasting tasks, where the purpose of the forecast is to meet or evade high or low values.

Simplified Expected Encounters

Definition. Given a set of real values $\mathrm{Y} = \{y_i\}_{i=1}^{n}$ and set of predicted values $\mathrm{P} = \{p_i\}_{i=1}^{n}$ we define EE(Y,P) as

$$\mathrm{EE(Y,P)} = \int_0^1 E(\lfloor r \cdot n \rfloor) dr \ , \tag{1}$$

where the function $E(k)$ is defined as

$$E(k) = \sum_{j=1}^{k} y_j \ . \tag{2}$$

Function $E(k)$ represents the cumulative sum of observed values y_j, where the values y_j from the set Y were sorted in ascending order using corresponding predicted values p_i as indices for sorting.

3 Anomaly and outlier detection methods

Outliers and their analysis are part of standard statistical data analysis. Before we tackle the problem of outlier detection, we must define what an outlier is. There exist multiple definitions of what is an outlier and what is not. There are also multiple ways how to classify outlier detection methods. The purpose of the following subsection is to look into outlier definitions and types of outliers. This section discusses these ideas and provides the basis of state-of-the-art outlier approaches. For simplicity, we will use the terms anomaly and outlier interchangeably.

3.1 Motivation

Outlier detection methods have numerous applications [31], ranging from credit-card fraud detection [32] to detecting people walking irregularly [33]. One of the major applications is finding anomalies in biological sensorial data, such as ECG [34], EEG [35], EMG [36] or actigraphy records [37]. An example of these would be heart arrhythmia in EEG, which is anomalous, and our task is to detect it. A common task is finding anomalies in biological image data, e.g., detecting malignant tumours in X-ray or MRI scans. Unusual symptoms, changes, or test results (such as blood results) may indicate potential health problems of a patient, could also be captured by anomaly detection algorithms. An example of this may be work [37] which proposes an algorithm for the classification of acute insomnia issues.

Another domain of anomaly detection might be in the robotics domain, where a security robot monitors pedestrian flows in a given hall. The application of anomaly detection methods is to detect the "suspicious" behaviour of humans, such as walking in the hall at night when the building should be closed, and no one should be there [23]. Such events may be considered for security reasons. In addition, the robot should be ready to change its spatio-temporal model and react accordingly if the model conflicts with a reality that seems anomalous. Analysis of rare events can serve the robot as additional information about the dynamics of the environment, which would help the robot plan its way through the environment.

The problem that concerns us is tied to the project called kdynakoupit.cz, run by Chronorobotics laboratory. A phone app called FreMEn Explorer, where people input the relative crowdedness at their courant position was developed to predict the occupancy of popular places [15]. A spatiotemporal model is then created to predict crowdedness at given locations over time. However, a few problems arise. All user's inputs are not homogeneous, and some measurements may not reflect reality well and may cause a problem for some methods. Thus, it would be a good idea to have a model that could detect anomalous and biased measurements from the context of time, place and measured values. In the same scenario, someone could deliberately sabotage the measurements by inputting completely wrong measurements. Another application in this scenario is not tied to the user's input

but to the nature of the data, where holidays, sales, or accidents may be considered an anomalous event which might throw the forecasting model off.

3.2 Prerequisites

3.2.1 Types of Outliers

In this thesis, we base our definition on Hawkins' definition [38] of outliers. It sounds: "An outlier is an observation that deviates so much from other observations as to arouse the suspicion that a different mechanism generated it." Hawkins' definition was also used by Breuning et al. in their work about a density-based method called Local Outlier Factor [39].

We can divide outliers into three significant categories - point, contextual and collective anomalies. Based on the reference set, we can also distinguish between local and global outliers. [40].

Point outliers are single datapoints that significantly deviate from other data points in the entire dataset. Point outliers are the simplest case of anomaly. They usually occur in categorical data or unordered sets of data.

Contextual outlier (also called conditional outlier) is a data point that differs from points with the same context. Datapoints labelled as outliers would not be labelled as outliers if they happened in a different context. As an example, the context can be temporal, spatial, or spatio-temporal. The temporal context is quite typical for time-series; e.g., considerable spikes in time-series are typical examples of contextual anomalies.

Collective outliers are subsequences that differ from the rest of the sequence. The sequences classified as anomalies would not be anomalies if they occurred alone. They can be found in time-series quite commonly as well. An example can be time-series, consisting of a sinusoid of 1Hz frequency, and suddenly the frequency of the sinusoid changes to 2 Hz for a short period but returns to 1Hz after some time. The subsequence with frequency 2Hz would be considered a collective outlier.

Local outlier is such a data point that is outlying with respect to a given subset or cluster in the dataset. The dataset may contain observations generated by a different mechanism, e.g., different probability distributions. The important thing is to identify the correct clusters in the data.

Global outlier is such a data point that is outlying with respect to the entire dataset. The only assumption made is that the same mechanism generated all data points, e.g., they come from the same distribution. Method for global outlier detection usually uses the entire dataset as a reference set which includes outliers.

3.2.2 Outlier detection approaches

There exist many ways how to classify anomaly detectors. Depending on the tweaks and specific use cases, some methods may not belong directly to any of the basic categories or may belong to multiple of them. The basic and the most straightforward way is to classify outlier detection methods by their output into classifiers and methods providing outlier scores [41]. They can also be divided into groups depending on whether they need labelled or scored training data to learn patterns for future predictions. These categories are called supervised and unsupervised. There also exist semi-supervised methods that combine both approaches. Anomaly detection can be used during data preprocessing or to detect novelties that do not fit the prediction. Specifically in time-series analysis, the anomaly detection is used almost exclusively for novelty detection [42].

Classifiers are one of the mentioned types. Outputs of these algorithms are usually binary labels, which indicates whether a given data point is an anomaly or not. It is a common practice to label anomalies as 1 and regular observations as 0. Multilabel classification can also be performed. In such a case, we have multiple different types of anomalies in the dataset and want to differentiate between them.

Outlier scores , on the contrary to classifiers, provide more information about the vector's "outlierness". Such methods usually estimate outlier scores as nonnegative real numbers. Very vaguely, the scores tell us how much a given point is outlying. The greater the anomaly score is, the more anomalous the point is. The outlier score is sometimes called the outlier factor. We do not need to classify points in some applications, and the anomaly score is sufficient or demanded. However, we would like to know what is an anomaly and what is not in most cases. Outlier scores can be thresholded to obtain binary labels. This means, if the point's outlier factor is greater than some threshold, then it is classified as an outlier and vice versa.

Supervised methods are algorithms that need labelled training datasets to train themselves [43]. An example can be decision trees, SVMs and some neural networks [41].

Unsupervised methods are algorithms that do not need labelled training datasets to train themselves, such as PCA [44], LOF [39], One Class SVM [45] and autoencoders [46, 47].

Anomaly detection on timeseries Anomaly detection methods in the time-series domain mostly rely on reconstruction error or forecasting residuals [48][49]. An example could be LSTM autoencoders [50] which are often used as baseline methods.

3.3 Selected methods

I chose few outlier detection methods that form the founding block of many anomaly detection concepts. They provide basic statistical, proximity-based interpretation of outliers.

Z-Score [51] is the most basic and simple method that can estimate outliers in univariate data. It is also the basic method in time-series novelty detection. Let us define Z-Score T_i of i-th observation x_i from set of observations $D = \{x_i\}_{i=1}^n$ as

$$T_i = \frac{x_i - \mu}{\sigma} \,, \tag{3}$$

where μ stands for the population mean of D and σ is the population standard deviation of D. Observations x_i, for which $|T_i| \geq T$ holds for some determined threshold T, are classified as outliers and vice versa. Samples x_i are assumed to be normally distributed. The famous rule, called 3-sigma rule, is based on the assumption that the observations x_i, for which $|T_i| < 3$ holds, lie within approximately $99,73\%$ two-sided symmetric confidence interval. The downside of the method lies in the estimation of parameters of underlying distribution when the distribution of population is unknown. The estimation is not robust because the mean and variance (or standard deviation) are easily influenced by outliers. A related and more robust tool to find outliers in univariate data is a boxplot. It was presented by Tukey in [52]. The boxplot is based on the quartile values of the data. The "box" is "centred" around the median (the second quartile) and has its lower and upper bounds given by the first and third quartiles. The interquartile range (IQR) is calculated as the difference between the third and first quartiles, IQR = q_3-q_1. Datapoint x is an anomaly if $x > q_3 + 1.5 \cdot \text{IQR}$ or $x < q_1 - 1.5 \cdot \text{IQR}$. Although the boxplot is more robust than the Z-Score, it does not provide an outlier score.

Mahalanobis distance [53] Mahalanobis distance $d_M(X|\mu, \Sigma)$ between random vector X and a given multivariate normal distribution $N(\mu, \Sigma)$ is defined as follows:

$$d_M(X|\mu, \Sigma)^2 = (X - \mu)^T \Sigma^{-1} (X - \mu) \,, \tag{4}$$

where μ and Σ are the distribution's mean and covariance matrix, respectively. This can be viewed as a multidimensional variation of Z-Score with the measure of distance similar to L_2 norm in curved space (when covariance matrix is idenity matrix, it becomes exactly L_2 norm). The square of Mahalanobis distance, $d_M(X|\mu, \Sigma)^2$, is connected with a chi-square distribution with n degrees of freedom, where n is a number of variables. In a simplified

way, $d_M(X|\mu,\Sigma)^2 \sim \chi^2_n$. That allows us to make a hypothesis on $d^2_M(X|\mu,\Sigma)$ and test it at chi-squared distribution's significance levels α, which gives Mahalanobis distance statistical interpretation. We can say that points with Mahalanobis distance greater than some threshold are outliers and vice versa, as in Z-Score's case.

The variable(s) X is not usually normally distributed, which leads to errors in the estimation of the parameters mean and covariance [54]. The estimated covariance matrix has to be regular, or the regularisation has to be applied. The parameters estimation is fundamental in the domain of anomaly detection because outliers can heavily influence the mean and covariance matrix and therefore affect the entire anomaly detection task [55]. Mahalanobis distance, by its definition, does not perform well in data generated by multimodal distribution or data forming multiple clusters. The method can be extended for multimodal data if the number of clusters is known or estimated using some of the clustering methods as well as the clusters themselves. However, it leads to a more complex pipeline with unpredictable quality of outlier detection. Mahalanobis distance is a kind of hybrid between statistical and proximity-based methods.

Local Outlier Factor is the first density-based outlier detection method presented by Breunig et al. [39]. Breunig tackled the problem with binary characteristics of outliers. Instead of assigning to the observation binary state of being an outlier or not, he came up with a scale named outlier factor, which characterizes the degree of outlying-ness of a given observation. The algorithm can detect outliers in multivariate data with multiple differently dense clusters but does not need prior information about clusters or their number. The method uses one hyperparameter k, which affects the cardinality of the neighbourhood of each observation. The choice of k affects its density by setting the boundary for the minimum vectors needed to form clusters. Vectors that lie inside the cluster or in dense areas have a local outlier factor approximately equal to 1, while vectors in sparse areas have an outlier factor greater than 1. Vectors with an outlier factor greater than 1 can be considered outliers in the case of binary classification.

The original paper uses a lot of make-up notation and functions that make the algorithm challenging to grasp. Vintrova tackles this problem in her doctoral thesis [56] and presents a general density-based algorithm for the local outlier detection task. After the LOF proposal, multiple ideas of how to improve Local Outlier Factors appeared, e.g., LOF', LOF", GridLOF proposed by Chiu et al. [57] or other works, such as LOCI [58], INFLO [59], LoOP [60].

3.4 Evaluation of outlier detection

This subsection is concerned with regularly used evaluation metrics in the outlier detection domain. Described metrics are based on the output of the confusion matrix, which needs binary classification as an input. To evaluate the method providing outlier scores,

we convert the given task into one or multiple binary classification tasks with a differently chosen threshold T.

Confusion Matrix is a two dimensional contingency table that allows visualisation of correctness of binary classification task, see Figure 1. The confusion matrix consists of four

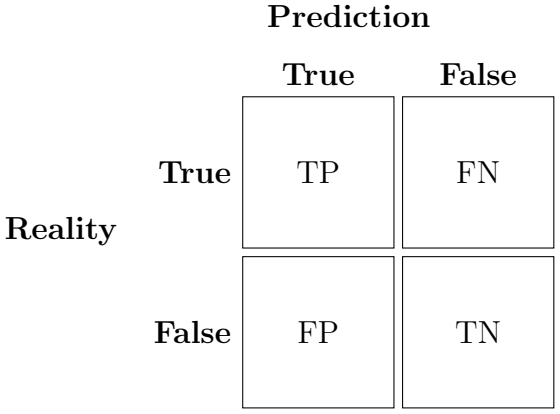


Figure 1: Visualization of confusion matrix

fields, True Positives, True Negatives, False Positives and False Negatives. For simplicity, let us refer to True Positives as TP, to True Negatives as TN, to False Positives as FP and to False Negatives as FN. We will use these abbreviations in upcoming definitions and terminology. There are different measures derived from the confusion matrix. True positive rate (TPR), sometimes also called Recall, is defined as

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5}$$

True negative rate (TNR) is defined as

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{6}$$

Precision is defined as

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{7}$$

More complex metrics, Matthews Correlation Coefficient (MCC) is defined as

$$\mathrm{MCC} = \frac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \,. \tag{8}$$

Matthews Correlation Coefficient, originally presented in [61], is just a discrete case of Pearson's Correlation Coefficient between variables X and Y, applied to the binary classification problem [62], where X is the actual label and Y is the predicted label.

Receiver Operating Characteristic Curve (ROC Curve) [63] is a graph, where False Positive Rate is plotted on the x-axis and True Positive Rate is plotted on the y-axis, while threshold T is variable. The ideal classifier is the one that has TRP equal 1 and FPR equal 0 for some value of T

Precision-Recall Curve (PR Curve) is a graph, where Recall is plotted on the x-axis and Precision is plotted on the y-axis with variable threshold T. The ideal classifier is the one with Recall and Precision equal 1 for some value T.

Area Under Curve (AUC) [63] is typically used in addition to the ROC curve. It provides the size of the area under the ROC curve. It summarizes the ROC curve as one number between 0 and 1, where 1 represents a perfect classifier.

4 Datasets

Outlier detection methods in this thesis are defined in such a way that they do not require labelled anomalies in the training data, which is supported by the fact that anomaly detection is quite commonly performed as an unsupervised task [64]. However, since we also want to evaluate outlier detection methods objectively, we need to have labels in the testing data. We are also looking for a dataset with strong periodic behaviour with a lack of trend, which is the basic assumption of chronorobotics forecasting methods.

I decided to test the hypotheses on synthetic periodic time-series data with synthetic outliers first, similar to the authors of [39, 45, 65, 66, 67] whom all used synthetic datasets in their works. Based on the outputs from the synthetic data tests, I will apply the methods to the real time-series from the FreMEn contra COVID database. The database consists of relative crowdedness measurements over multiple places in Czechia.

All tested time-series in this thesis have the same structure of time-dependent variable derived from the real datasets. The values can acquire integer values between zero and five, where each of the values has a qualitative meaning:

- 0 - Closed,
- 1 - Empty,
- 2 - Low Traffic,
- 3 - Medium Traffic,
- 4 - High Traffic,
- 5 - Full, Crowded.

Although these qualitative values lack the precision compared to the number of people at the place, it has its advantages. First of all, it is effortless to estimate the value during measurement. Such measurement also does not violate the usual requests of the owners of measured places, who find the information about the exact number of people in their place private. The values are also comparable between differently large places, as the meaning of the values is "crowdedness relative to the size of the place".

4.1 Real datasets and possible scenario

The information system of the project FreMEn contra COVID was finished during the writing stage of my thesis. The database consisted of a relatively small amount of data. As the whole system is quite complex and generalises the information gathered from different places, the time-series from individual places were not of the quality suitable for my experiments. I decided to provide the system with my own measurements over seven places in proximity of the university building. The measured values of relative crowdness are used in the last experiment.

Measured places

1. Albert - Karlovo nám. 15, 120 00 Nové Město, Praha
2. DM - Karlovo nám. 292/14, 120 00 Nové Město, Praha
3. Billa - Atrium, Karlovo nám. 2097/10, 120 00 Praha
4. Dr. Max - Karlovo nám. 313/8, 120 00 Nové Město, Praha
5. Costa Coffee - Karlovo nám. 8, 120 00 Nové Město, Praha
6. Bistro - Václavská pasáž, 120 00 Nové Město, Praha
7. Svatováclavská cukrárna - Václavská pasáž, 120 00 Nové Město, Praha

The training data were gathered during three weeks of systematic measuring. I measure at random times of the days, usually ten times a day. I did not measure every day. Some days I measured only a few times. Every training time-series consists of approximately 150 measurements.

The test data were gathered during one day. Every place was measured every thirty minutes with a small deviation as possible. The measurements also included the exact number of people for the further and more complex experiments. Every test time-series consists of 49 measurements. As the purpose of the data is to predict the relative crowdedness of the places and my thesis concerns with outlier detection, I needed to include and label synthetic outliers into the test data, see Section 4.3.

4.2 Synthetic datasets

Various synthetic time series scenarios were designed to test anomaly detectors and their characteristics to the full extent. This subsection describes each of the scenarios as well as a general approach to generate synthetic datasets. Each scenario defines a generating function or multiple generating functions from which we sample values at random times. Gaussian noise is added to the sampled values and rounded to the nearest integer after that. Each scenario consists of hundreds of time-series generated with the number of "measurements" in the range between 60 and 240 during three weeks. Every test dataset is generated during the "seventh" week, i.e., three weeks after the training dataset. Every test time-series consists of 2016 rounded values obtained every 5 minutes. 10% of the values were changed and labelled. They serve as outliers. Every outlier is an integer between 0 and 5 with the difference from the original value at least 2.

Weekend scenario The first scenario is called the Weekend scenario. In this scenario, the generating function evinces daily and weekly periodicities. Peaks of values happen during working day noons, see Figure 2.

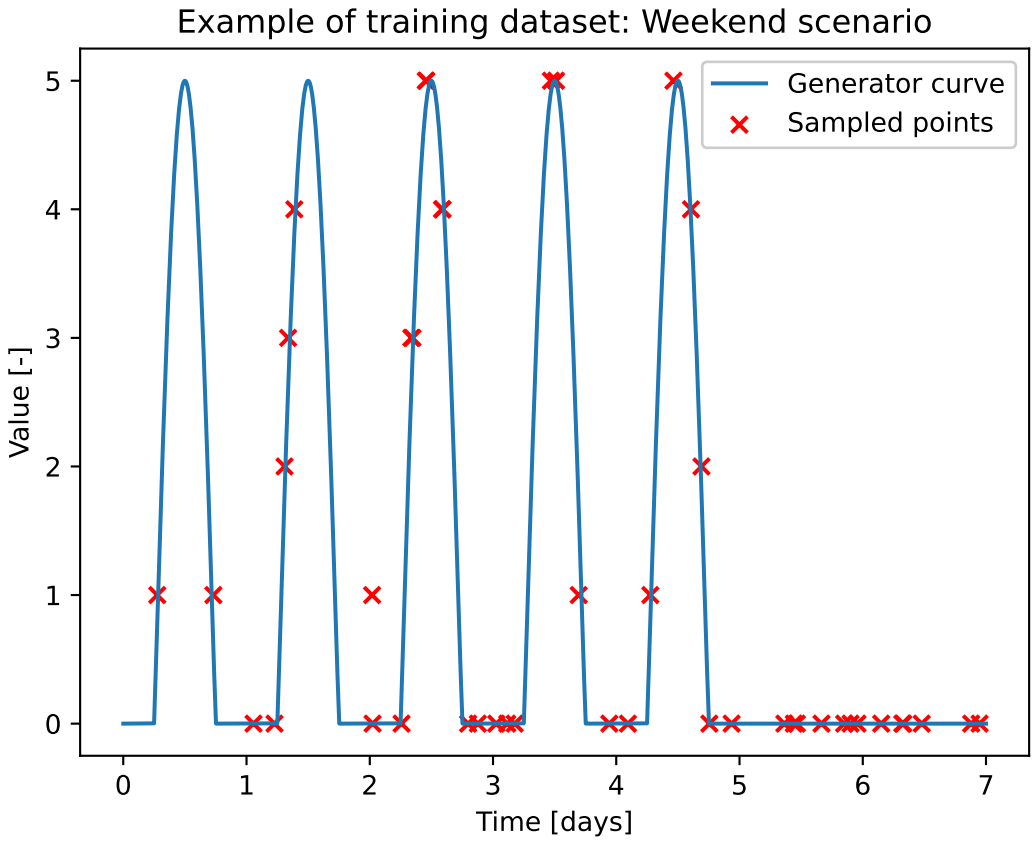


Figure 2: Example of 1 week of training data with 40 sampled points in Weekend scenario, including graph of generating function.

Lunch scenario The second scenario is called the Lunch scenario and is similar to Weekend scenario. However, peaks happen twice a working day, once before and once after lunch, except with weekend, where peak exhibits at noon, see Figure 3.

Bimodal scenario The third scenario is called the Bimodal scenario. It was designed to test the ability of methods to detect outliers in time-dependent variables with the multimodal distribution. The data is generated using two processes, where each has a periodicity of 2 days, and their mutual phase shift corresponds to 1 day. After the random sampling is performed, the value is gathered randomly from one of generating functions, see Figure 4.

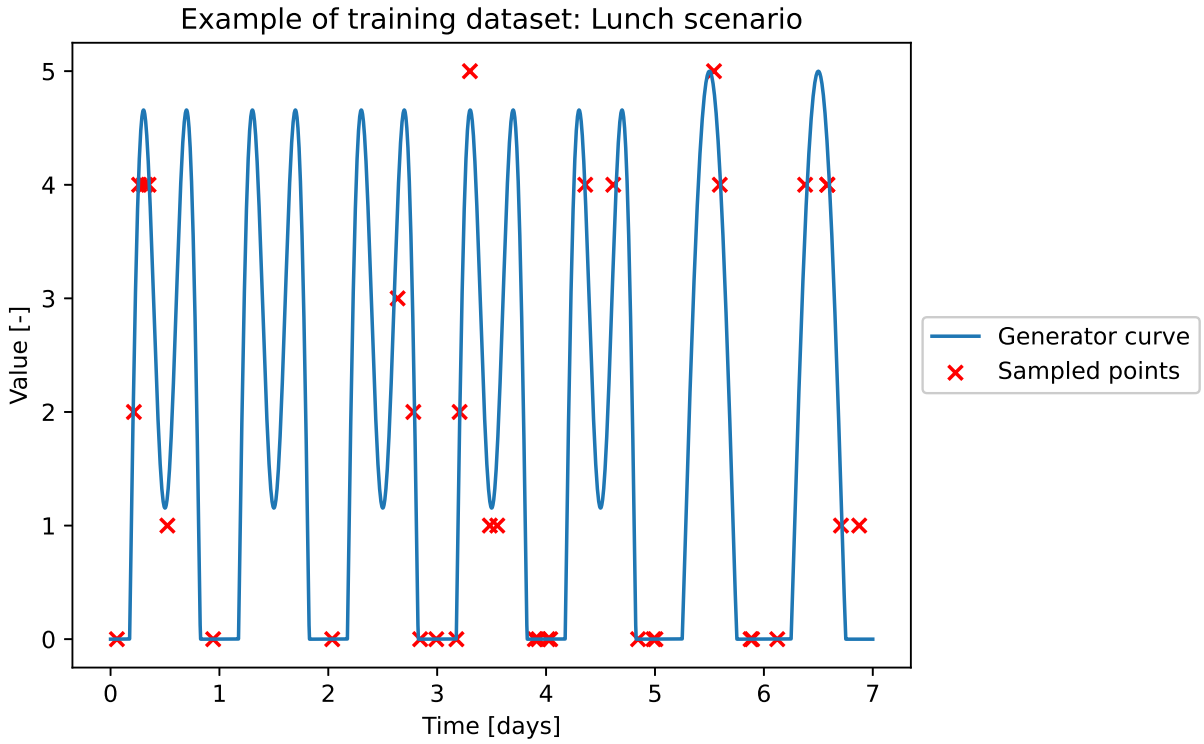


Figure 3: Example of 1 week of training data with 40 sampled points in Lunch scenario, including graph of generating function.

4.3 Synthetic outliers

In both synthetic and real time-series, we need to include labelled outliers. The real data, by its nature, do not consist of known outliers that can be labelled. We expect that natural phenomena generated anything that happened during the test day.

The timestamps with the outlying values were chosen as a random subset consisting of 10% of all timestamps in every test time-series. The distance between the value of generating function at the timestamp and the outlying value was between 2 and 5, but every outlying value was integer between 0 and 5. In the Bimodal scenario, the only possible difference suitable for the time-series was 2, because the largest distance between functions was 4, and the largest distance to the maximum possible values from both functions together was 3 (but only in a minimal number of timestamps). The different shifts of values were chosen uniformly, i.e., the same amount covered every distance and direction.

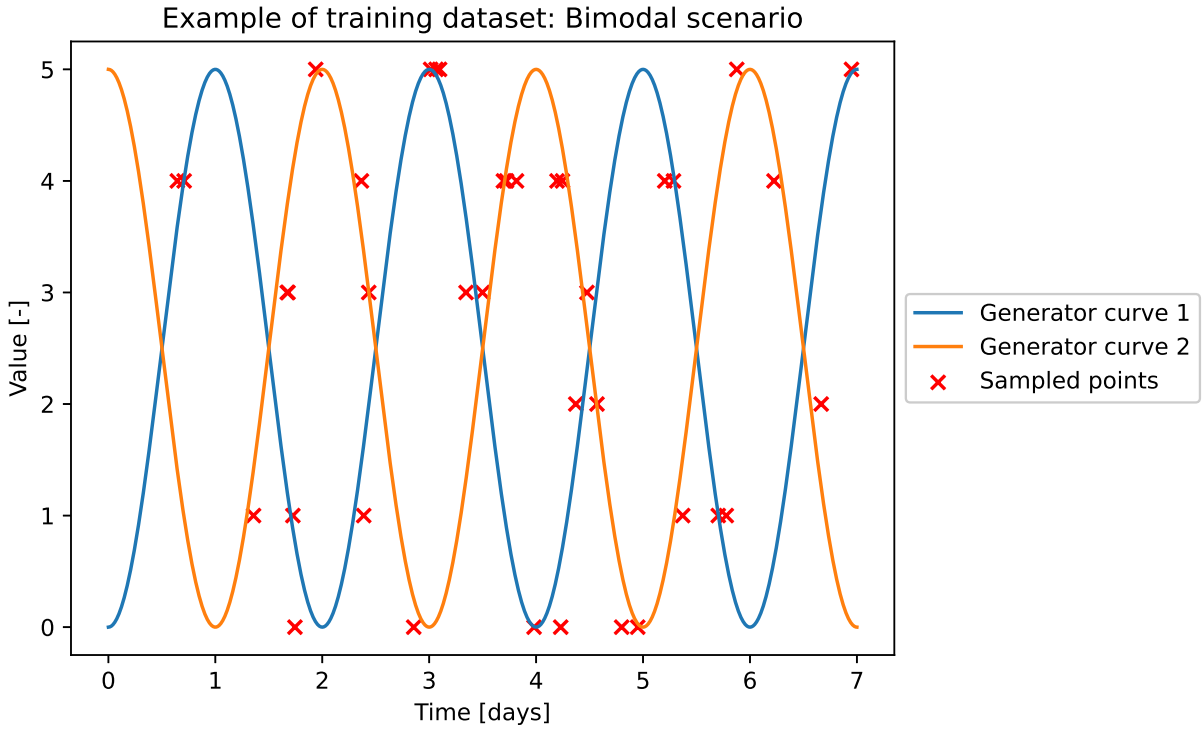


Figure 4: Example of 1 week of training data with 40 sampled points in Bimodal scenario, including graph of generating functions.

5 Methods in testing environment

This section revisits methods used in follow-up experiments and describes the testing environment. All experiments are performed in the docker [68]. It is possible to include new methods and new datasets into the experiments. The methods are implemented using python3 [69] with standard libraries. Machine learning and outlier detection methods are implemented using scikit-learn library [70]. Some heavy processing functions of FreMEn and WHyTe are implemented using cython [71].

Note The experimental environment follows up Kubis's work [14], who designed automated evaluation tool. Special thanks go to Zdenek Rozsypalek, who implemented and set up the docker. In terms of the tool, my work extends Kubis's benchmarking tool with the interface for anomaly detection methods, datasets, experiments, evaluation metrics, and visualisation.

5.1 Forecasting methods

All implemented forecasting methods follow standard setup - they include following functions:

- **fit** takes 2-dimensional array of times and dependant varibale values and outputs leaned model.
- **predict** takes 1-dimensional array of times and outputs prediction of dependant variable values.

FreMEn The datasets for FreMEn are discretised into six cells (with indexes 0, 1, 2, 3, 4, 5), where each cell has its binary dataset. The cell indexes j represent the original value assigned to the cell. Each binary dataset poses as an indicator of whether the state at a given cell occurred.

The reconstruction of regression curve is implemented as in the original paper, which proposes to take the argument of maxima of cell state probabilities:

$$y(t) = \underset{j}{\operatorname{argmax}}\, p_j(t) \ , \tag{9}$$

where s_i is values assigned to cell and $p_i(t)$ is probability of an event occurence at j-th cell. Probabilities $p_i(t)$ are computed for discretized dataset according to original paper using Fourier transfrom and Fourier series.

WHyTe Warped Hypertime was implemented according to original paper [26].

Prophet We used the implementation of Prophet from the fbprophet library. The model is automatised and does not require any parameter fitting. The only wrapper for benchmarking tool interface was built. The wrapping includes converting timestamps into pandas library [72] Dataframe format, which is required at Prophet's input.

Historical models Historical models build their model on a period of P, divided into multiple bins depending on each bin width and the overall number of bins. The prediction at time t is calculated as an average of training dependent variable values that "fall" into the same bin as (t *modulo* P). We use three modifications of historical models:

- **weekly model** which is a historical model with the period of one week.
- **daily model** which presents a historical model with the period of one day.
- **mean model** which predicts an average, calculated over all training values of the dependent variable.

The number of equally spaced bins for Weekly and Daily models was chosen as $\lceil\sqrt{n}\rceil$, where n represents the number of training data samples.

Model name	Description
WHyTe	Warped Hypertime predictor
FreMEn	Frequency Map Enhancement predictor
Prophet	See paragraph about Prophet
Weekly	Historical model with weekly period
Daily	Historical model with weekly period
Mean	Historical model over entire training data

Table 1: Table of used forecasting methods

5.2 Anomaly detection methods

For this thesis, we suggest multiple anomaly detection methods. All them follow same template, which includes implementing following functions:

- **fit** takes 2-dimensional array of times and dependent variable values as input and outputs trained model.
- **predict_scores** takes 2-dimensional array of times and dependent variable values as input and returns outlier score assigned to given datapoints,
- **predict** takes 2-dimensional array of times and dependent variable values as input and returns outlier binary label (1 means outlier, 0 means inlier).

5.2.1 Forcasting methods with default outlier detection

FreMEn Outlier factor assigned observation that "falls" into j-th bin at time t is calculated as

$$\mathrm{OF}(j|t) = 1 - p_j(t) \ , \tag{10}$$

where $p_j(t)$ is probability of occurence of observation, which corresponds to j-th cell at time t. If $\mathrm{OF}(j|t) > 1 - \alpha$, for some given α, (t, j) is classified as an outlier. Note that this works only for discrete variables.

Prophet The default outlier detector in Prophet uses an asymmetric confidence interval around the forecasting function. I use its confidence interval boundaries similar to Z-Score's standard deviation.

5.2.2 Regressive outliers methods

Baseline methods for outlier detection in timeries is analysing residuals(error) between predicted value and actual value:

$$\mathrm{error} = f(\mathrm{target} - \mathrm{prediction}), \tag{11}$$

where function f may represent for example absolute value or square. However, in our case, we want to use signer error because of the nature of data. Therefore, function f is identity function, i.e., $f(x) = x$. At first, general training process of our regressive method is introduced. Anomaly detection phase is dectibed after that.

Learning phase

1. Make prediction at given times t using given forecasting method.
2. Calculate errors between target and prediction.
3. Analyse calculated errors to build anomaly models.
4. Set threshold T as a boundary for outlierness.

Anomaly detection phase

1. Make prediction at give times t using given forecasting method.
2. Calculate errors between target values and predicted values.
3. Compare errors with set threshold T.

Choosing forecasting and error analysis methods

- **Forecasting methods** - arbitrary forecasting method, that predicts single one dimensional value x at time t, can be applied. We use all methods described in subsection 5.1.
- **Error analysis/outlier detection methods** - arbitrary one dimensional anomaly detection method might be used. Analysis can be performed on raw error or after normalising errors by Z-Score. We use Z-score, which normalises errors so that they are centred around zero and have unit variance. Also, LOF [39] is applied to analyse errors in our case.

5.2.3 Hypertime Transform based methods

We propose a new type of anomaly detection methods based on Warped Hypertime Transformation [26], which we will refer to as HyT. The structure of these outlier detectors has unusual learning and anomaly detection phase.

Learning phase

1. Find Hypertime Transform parameters (done automatically by HyT method).
2. Perform Hypertime space expansion using HyT.
3. Learn outlier structure over expanded Hypertime space using standard anomaly detection method.

Anomaly detection phase

1. Perform Hypertime space expansion using HyT.
2. Predict outlier scores over expanded Hypertime space.

Almost any arbitrary anomaly detection method that estimates outlier scores in multivariate data can be applied to the expanded Hypertime space.

List of specific anomaly detection methods

- **LOF over HyT** Local Outlier Factor [39] estimates density over expanded Hypertime space. The points with low density (high LOF) are labeled as outliers.
- **OC-SVM over HyT** One-Class SVM [45] with RBF kernel is used to find the hyperplane that separates points based on their desinties the best.
- **Mahalanobis distance over HyT** Mahalanobis distance described in 3 in applied on expanded Hypertime space.

Model name	**Description**
FreMEn Detector	Default FreMEn outlier detector
Prophet Detector	Deafult Prophet outlier detector
HyT+LOF	LOF over Hypertime space
HyT+OC-SVM	OC-SVM over Hypertime space
HyT+MD	Mahalanobis distance over Hypertime space
Z-Score+[arbitrary forecasting method]	Z-Score over forecasting erros
LOF+[arbitrary forecasting method]	LOF over forecasting errors

Table 2: List of anomaly detection methods

6 Experiments

We designed different experiments to assess the ability of anomaly detectors. Each experiments' structure is defined as a combination of used datasets, anomaly detection methods, evaluation criteria and visual output.

Methods All the methods described in the subsection about anomaly detectors are used in all of the presented experiments. This includes regressive outlier detection methods and Hypertime based anomaly detection methods.

6.1 Experiment ROC

This experiment was designed to compare different approaches to anomaly detection methods in various scenarios. The Receiver Operating Curve with the area under its curve is used to evaluate the overall ability of tested anomaly detectors over multiple thresholds. Along the ROC curve, the Precision-Recall curve is for comparison and as a consistency check. In addition to the PR curve, a similar metric to the Area Under PR Curve called the Average Precision metric is chosen. It implements average precision over all given thresholds.

Datasets

- **Training datasets** consist of 140 randomly generated datasets [ref datasety] over 3 different scenarios (Weekend, Lunch and Bimodal) with 7 different number of measurements in training data - 60, 90, 120, 150, 180, 210, 240.
- **Testing datasets** contain one dataset for each scenario (Weekend, Lunch and Bimodal) with 10% arficially generated outliers. Which means we have 3 testing sets, each with 2016 regular measurements over 1 week, where 202 measurements are outliers.

Metrics Receiver Operation Characteristic and Precision-Recall curve, along with Area Under Curve (for ROC) and Average Precision (for PR), are used in this experiment to compare how well models are classifying without assuming any specific boundary for the outlierness.

Process of running one scenario

1. Generate all datasets for a given scenario, which includes 20 batches of training datasets, where each batch has 7 datasets according to the number of measurements.

2. Train anomaly detection methods.
3. Estimate outlier scores in testing data using trained anomaly methods.
4. Calculate mean ROC and mean PR curves over all batches and the number of measurements for each method.
5. Visualize results.

The described process is run for each scenario.

6.1.1 Experiment ROC - Results

The figures show that for almost every method, its corresponding AUC was greater than its AP score. This fact might mean that the ROC curve overestimates either detector's prediction, or the PR curve underestimates the prediction. We test this observation further in the experiments. However, the relative order between detectors mainly stayed the same while comparing ROC and PR curves.

The output of this experiment serves more like a visual guide into how good are detectors between each other. Curves summarizing the Bimodal scenario show that effectively no method, except Chronorobotics methods, can work with this type of data.

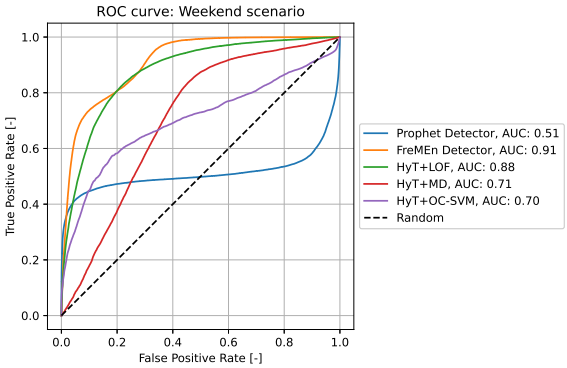


Figure 5: ROC curve in Weekend scenario - Chronorobotics methods + Prophet

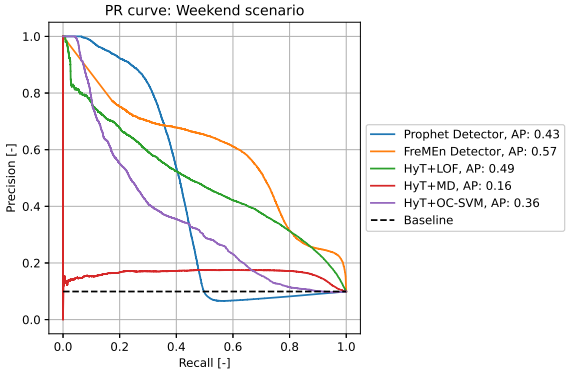


Figure 6: PR curve in Weekend scenario - Chronorobotics methods + Prophet

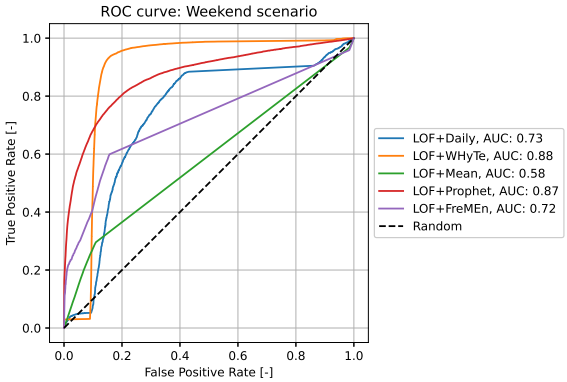


Figure 7: ROC curve in Weekend scenario - Regressive outlier methods with LOF

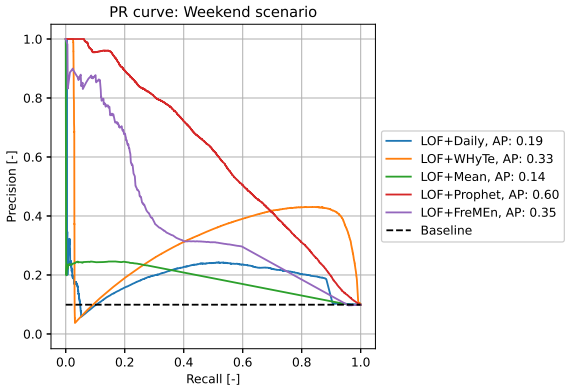


Figure 8: PR curve in Weekend scenario - Regressive outlier methods with LOF

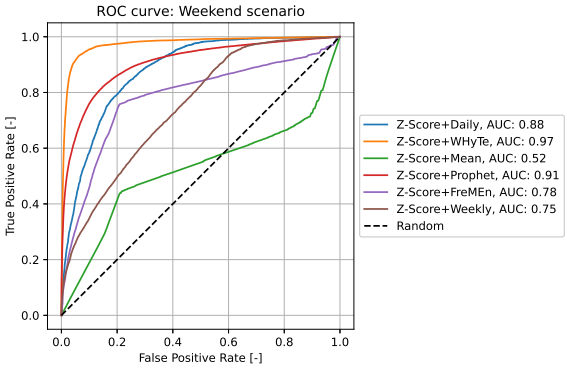


Figure 9: ROC curve in Weekend scenario - Regressive outlier methods with Z-Score

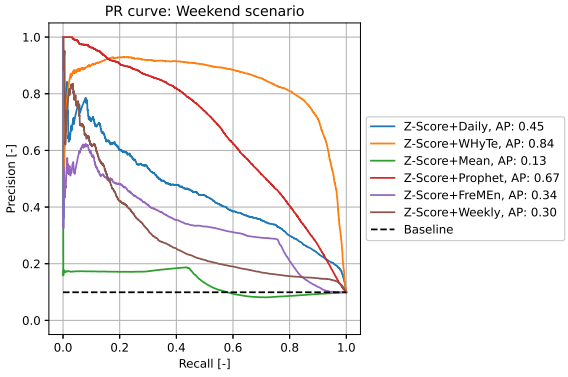


Figure 10: PR curve in Weekend scenario - Regressive outlier methods with Z-score

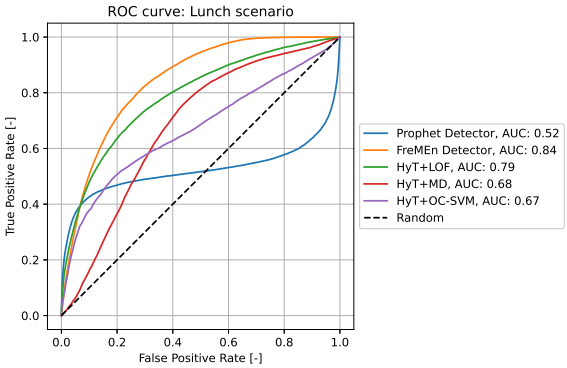


Figure 11: ROC curve in Lunch scenario - Chronorobotics methods and Prophet

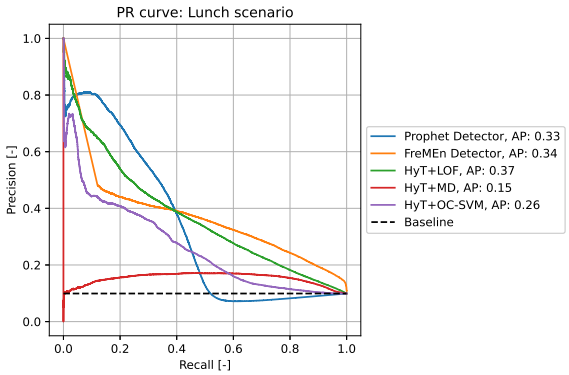


Figure 12: PR curve in lunch scenario - Chronorobotics methods and Prophet

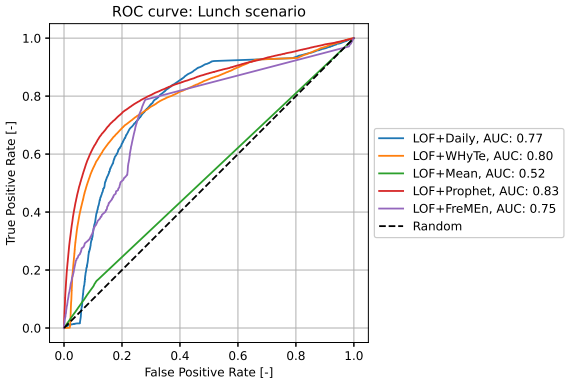


Figure 13: ROC curve in Lunch scenario - Regressive outlier methods with LOF

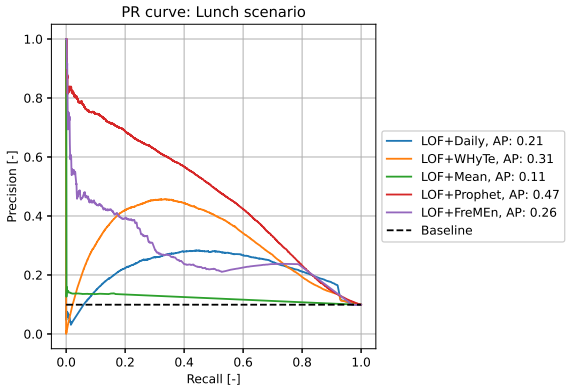


Figure 14: PR curve in Lunch scenario - Regressive outlier methods with LOF

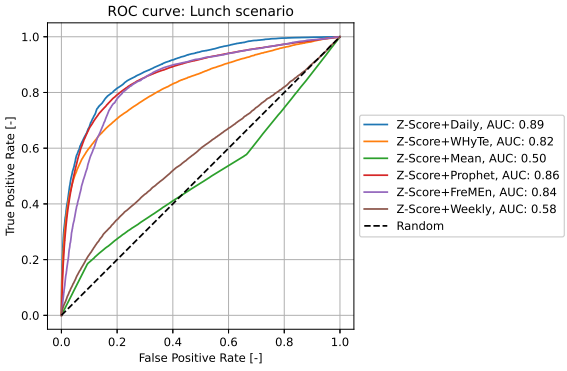


Figure 15: ROC curve in Lunch scenario - Regressive outlier methods with Z-Score

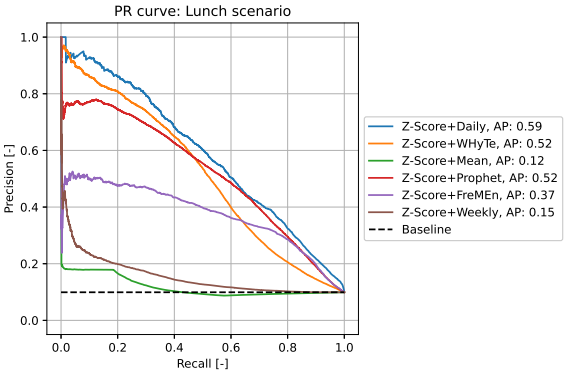


Figure 16: PR curve in Lunch scenario - Regressive outlier methods with Z-Score

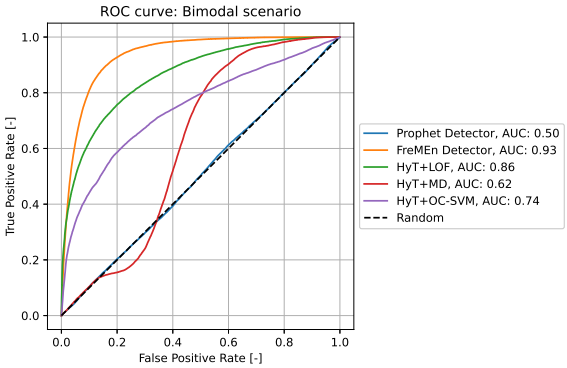


Figure 17: ROC curve in Bimodal scenario - Chronorobotics methods and Prophet

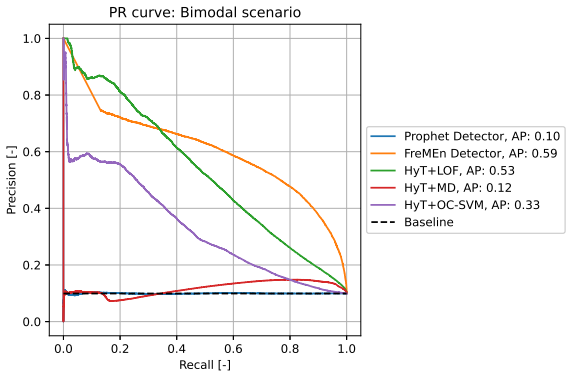


Figure 18: PR curve in bimodal scenario - Chronorobotics methods and Prophet

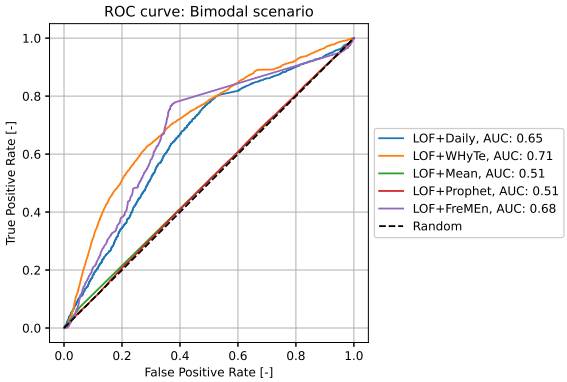


Figure 19: ROC curve in Bimodal scenario - Regressive outlier methods with LOF

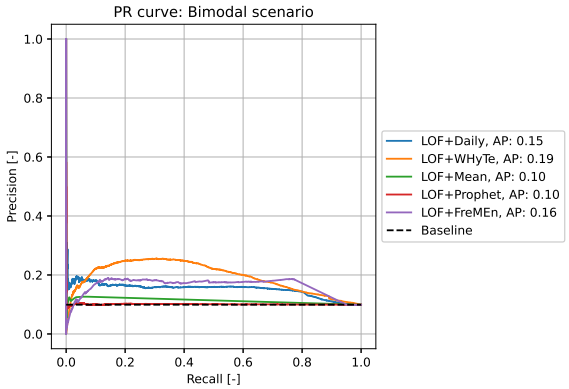


Figure 20: PR curve in Bimodal scenario - Regressive outlier methods with LOF

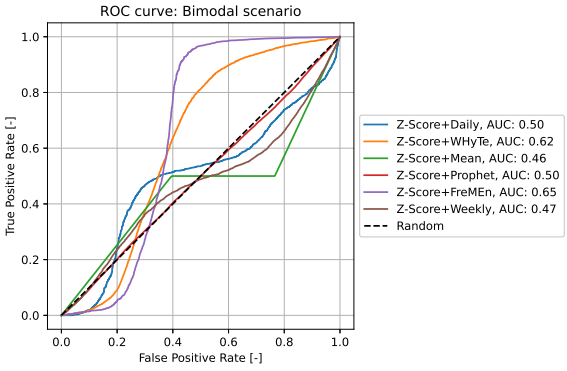


Figure 21: ROC curve in Bimodal scnario - Regressive outlier methods with Z-Score

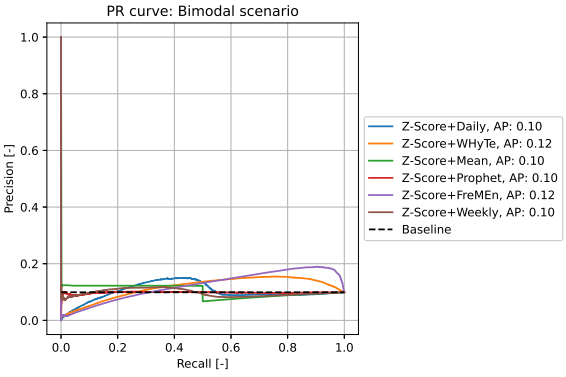


Figure 22: PR curve in Bimodal scenario - Regressive outlier methods with Z-Score

6.2 Threshold calibration experiment

This experiment was designed to choose an optimal threshold that converts outlier scores to binary labels and serves as a decision boundary between inliers and outliers.

Datasets

- **Training datasets** contains 100 random combinations of scenario and number of measurements and generate all corresponding datasets
- **Testing datasets** contain one dataset for each scenario (Weekend, Lunch and Bimodal) with synthetic outliers. This means we have 3 testing sets, each with 2016 regular measurements with 202 outliers over 1 week.

Criteria Criteria for optimal threshold is chosen as argument of maxima Matthews Correlation Coefficients in MCC curve. This curve represents MCC scores for every distinct possible threshold.

process of running experiment

1. Choose 100 random combinations of scenario and the number of measurements and generate all corresponding datasets.
2. Train anomaly detection methods.
3. Estimate outlier scores in testing data using trained anomaly detection methods.
4. Calculate mean MCC over all generated datasets for each method.
5. Find optimal threshold for each method.
6. Visualize results as MCC curve for each method.

6.2.1 Threshold calibration experiment - Results

One of the conclusions is that LOF in regressive outlier methods does not provide any more advantage over Z-Score. Moreover, Z-Score is more computationally efficient and, on average, even more accurate in MCC.

The experiment results in distribution of detector's MCCs over possible thresholds. It provides us with optimal threshold and also gives us an idea about its stability by looking at changes and slopes in surroundings of the optimum. Some of the specific MCC curves and chosen thresholds do not make sense. An example of this may be the MCC curve for LOF+FreMEn and corresponding optimal threshold, which was set to approximately 10^{10}. We believe this was caused by the numerical instability of LOF on sparse discrete data.

Anomaly detector	Threshold
Prophet Detector	2.6
FreMEn Detector	0.9
HyT+LOF	1.1
HyT+MD	14
HyT+OC-SVM	3.1
LOF+Daily	1.2
LOF+WHyTe	3.2
LOF+Mean	1.0
LOF+Prophet	4.1
LOF+FreMEn	$\approx 10^{10}$
Z-Score+Daily	1.6
Z-Score+WHyTe	4.2
Z-Score+Mean	0.8
Z-Score+Prophet	3.1
Z-Score+FreMEn	0.6
Z-Score+Weekly	2.7

Table 3: Optimal thresholds for anomaly detectors according to Calibration experiment

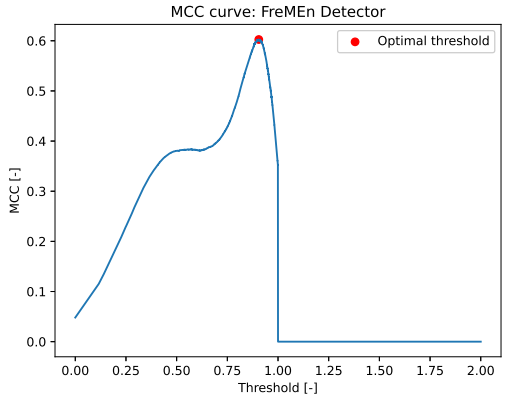


Figure 23: MCC curve for FreMEn Detector in Calibration experiment

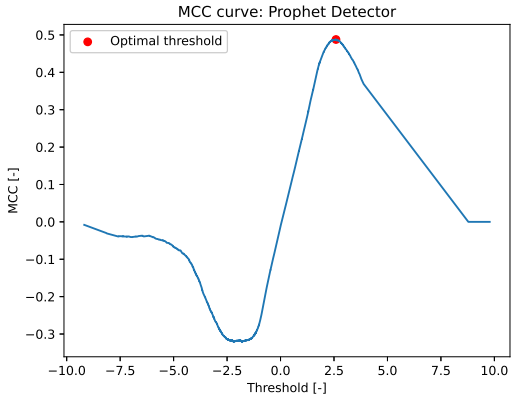


Figure 24: MCC curve for Prophet Detector in Calibration experiment

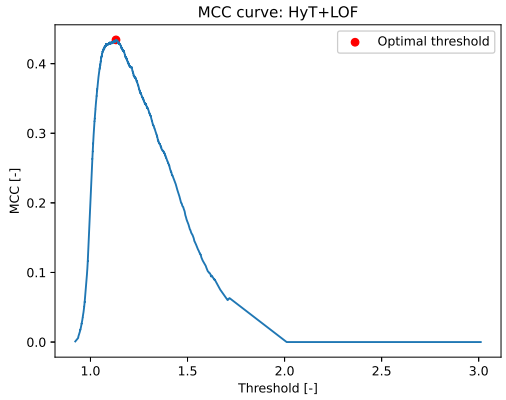


Figure 25: MCC curve for HyT+LOF in Calibration experiment

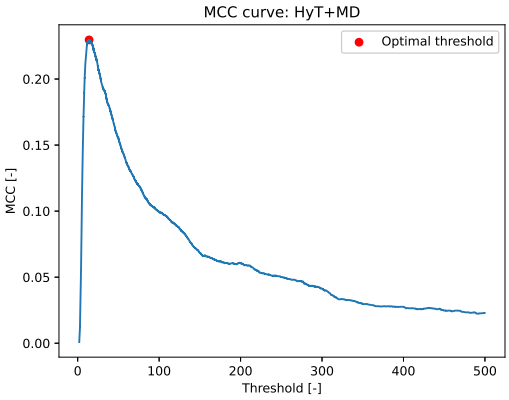


Figure 26: MCC curve for HyT+MD in Calibration experiment

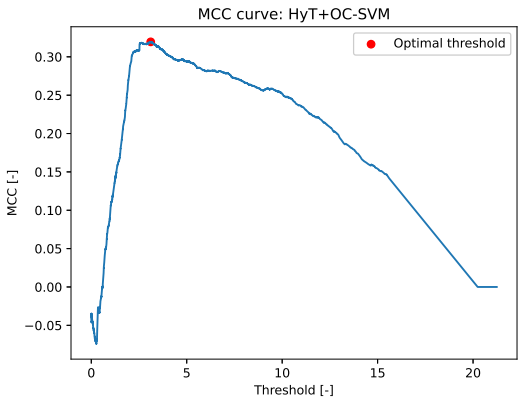


Figure 27: MCC curve for HyT+OC-SVM in Calibration experiment

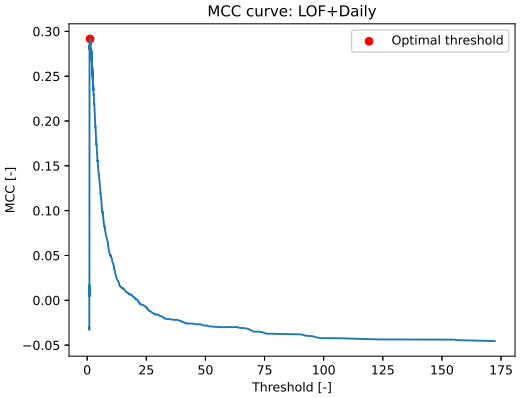


Figure 28: MCC curve for LOF+Daily in Calibration experiment

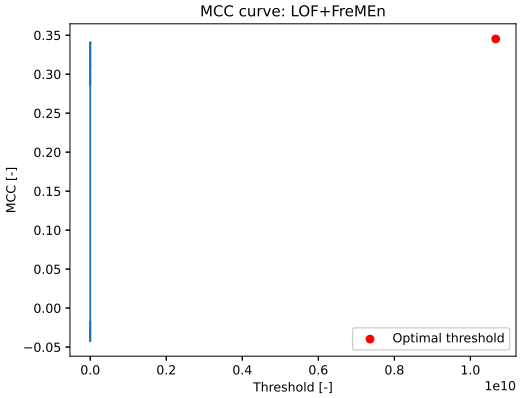


Figure 29: MCC curve for LOF+FreMEn in Calibration experiment

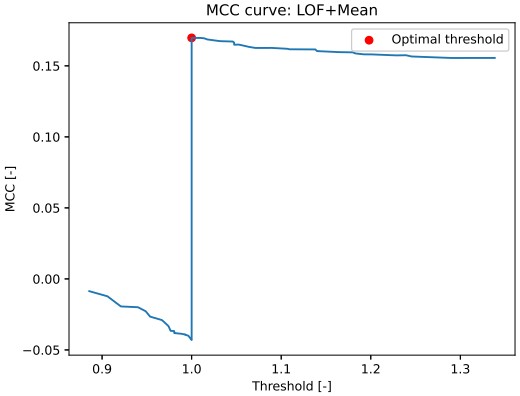


Figure 30: MCC curve for LOF+Mean in Calibration experiment

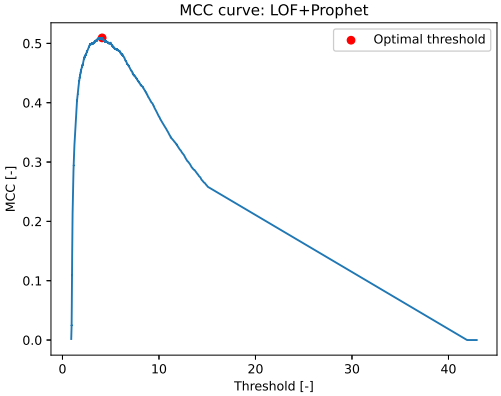


Figure 31: MCC curve for LOF+Prophet in Calibration experiment

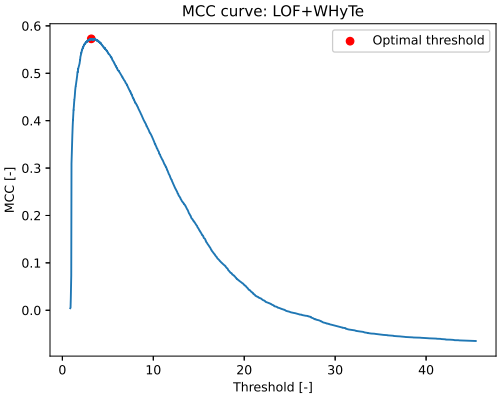


Figure 32: MCC curve for LOF+WHyTe in Calibration experiment

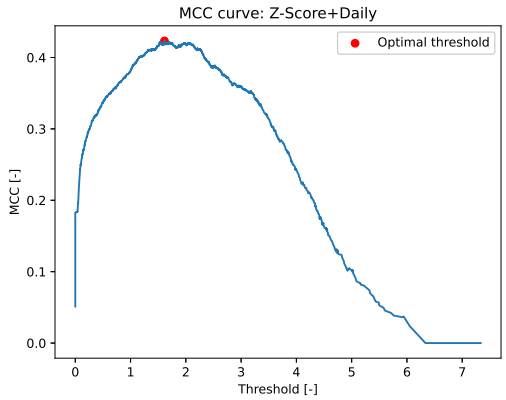


Figure 33: MCC curve for Z-Score+Daily in Calibration experiment

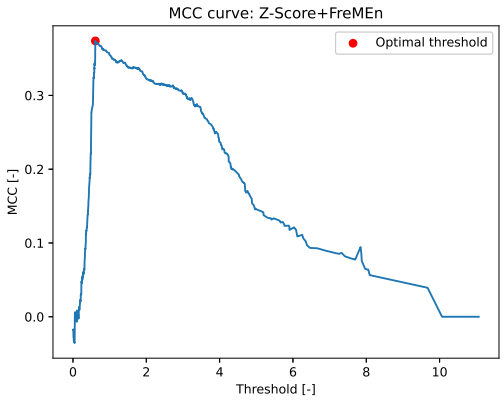


Figure 34: MCC curve for Z-Score+FreMEn in Calibration experiment

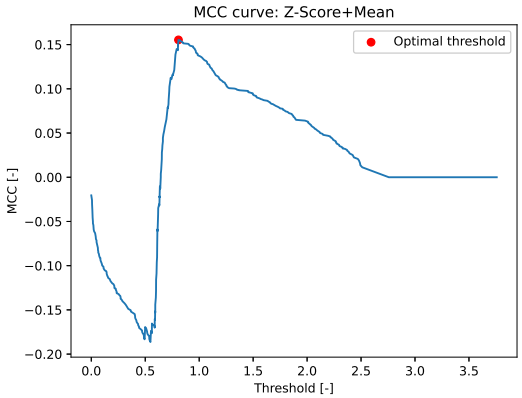


Figure 35: MCC curve for Z-Score+mean in Calibration experiment

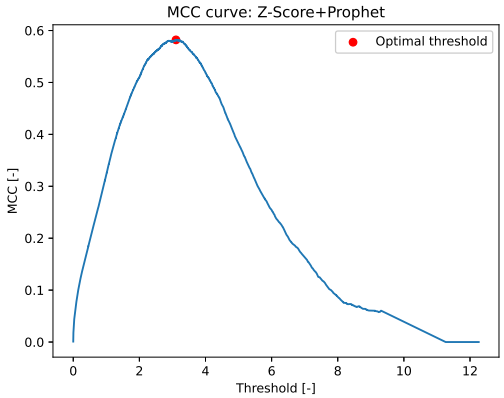


Figure 36: MCC curve for Z-Score+Prophet in Calibration experiment

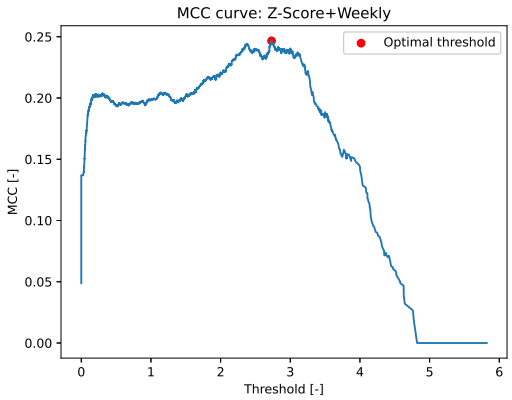


Figure 37: MCC curve for Z-Score+Weekly in Calibration experiment

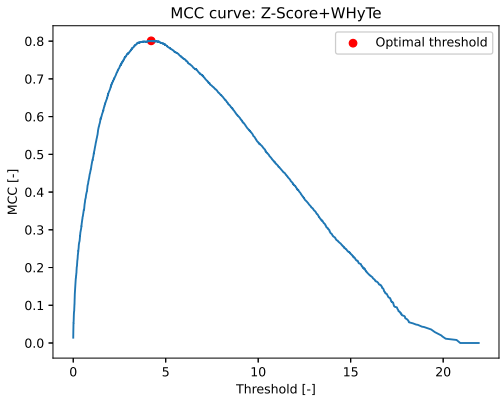


Figure 38: MCC curve for Z-Score+WHyTe in Calibration experiment

6.3 Experiment MCC

This experiment was designed to evaluate multiple methods with an already chosen optimal threshold on one dataset or to evaluate one method with the chosen threshold on multiple datasets. The results also provide info on how the number of measurements in training data affects the model's detective abilities.

Datasets

- **Training datasets** consist of 140 randomly generated datasets [ref datasety] over 3 different scenarios (Weekend, Lunch and Bimodal) with 7 different number of measurements in training data - 60, 90, 120, 150, 180, 210, 240.
- **Testing datasets** contain one dataset for each scenario (Weekend, Lunch and Bimodal) with 10% artificially generated outliers. Which means we have 3 testing sets, each with 2016 regular measurements with 202 outliers over 1 week.

Metrics Matthews Correlation Coefficient is an appropriate metric to evaluate the anomaly detector's ability to differentiate between outliers and inliers.

Process of running one scenario

1. Generate all datasets for a given scenario, which includes 20 batches of training datasets, where each batch has 7 datasets according to the number of measurements.
2. Train anomaly detection method on each combination.
3. Estimate outlier scores in testing data using trained anomaly methods.
4. Calculate the mean and standard deviation of MCC for each combination of method and number of measurements.
5. Visualize results in the table.

The described process is running for each scenario.

6.3.1 Experiment MCC - Results

Results - Weekend scenario The table 4 shows that method with the overall highest average MCC in the Weekend scenario is WHyTE+Z-Score. It scored stably in the first place in terms of average MCC. However, its standard deviation of MCC scores was among the highest in all number of measurements, except the last one (240 measurements), where the standard deviation lowered significantly. The second method in terms of average MCC

was Prophet+Z-Score. Even LOF+WHyte and LOF+Prohpet performed exceptionally well compared to others. Regressive methods with WHyTe and Prophet both did not suffer from a lower amount of data. FreMEn detector's and HyT+LOF's performance was average.

Results - Lunch scenario According to table 5, WHyTe did not perform that well in the Lunch scenario when having a smaller volume of training data. WHyTe's standard deviation across all datasets was still the highest one among other methods. Prophet dominated in most of the cases; his standard deviation was lower in general.

Results - Bimodal scenario On the contrary to previous scenarios, the FreMEn detector and HyT+LOF had the best overall MCC score in the Bimodal scenario 6 which was expected due to characteristics of these methods to multimodal model distributions. Others, except HyT+OC-SVM, predicted outliers similarly to the random classifier.

The experiment confirms that the-state-of-the-art anomaly detection methods cannot predict outliers in multimodal data. On the contrary, methods using the Chronorobotics approach seem to work quite well in this case 6. This fact confirms that methods developed in the spirit of Chronobotics ideology can model multimodal phenomena in time.

Another observation from conducted experiments in the different scenario is that the amount of training data limits anomaly detector's performance, especially methods based on density. This fact applies to all scenarios. A lower volume of training data resulted in lower MCC, which was expected.

	60	90	120	150	180	210	240
Prophet Detector	0.44 ± 0.10	0.47 ± 0.10	0.49 ± 0.08	0.52 ± 0.05	0.51 ± 0.05	0.52 ± 0.04	0.52 ± 0.06
FreMEn Detector	0.43 ± 0.05	0.49 ± 0.07	0.57 ± 0.07	0.63 ± 0.10	0.63 ± 0.08	0.68 ± 0.08	0.71 ± 0.07
HyT+LOF	0.19 ± 0.07	0.33 ± 0.08	0.45 ± 0.08	0.49 ± 0.06	0.53 ± 0.05	0.54 ± 0.04	0.57 ± 0.06
HyT+MD	0.20 ± 0.08	0.22 ± 0.09	0.22 ± 0.10	0.25 ± 0.10	0.26 ± 0.08	0.23 ± 0.10	0.24 ± 0.09
HyT+OC-SVM	0.29 ± 0.09	0.38 ± 0.09	0.32 ± 0.11	0.34 ± 0.08	0.34 ± 0.07	0.35 ± 0.05	0.33 ± 0.08
LOF+Daily	0.27 ± 0.06	0.30 ± 0.05	0.33 ± 0.08	0.30 ± 0.08	0.30 ± 0.08	0.26 ± 0.07	0.28 ± 0.06
LOF+WHyTe	0.49 ± 0.14	0.52 ± 0.17	0.55 ± 0.09	0.53 ± 0.15	0.62 ± 0.11	0.67 ± 0.08	0.69 ± 0.09
LOF+Mean	0.26 ± 0.03	0.21 ± 0.06	0.13 ± 0.03	0.12 ± 0.04	0.14 ± 0.04	0.16 ± 0.06	0.12 ± 0.06
LOF+Prophet	0.52 ± 0.12	0.54 ± 0.13	0.59 ± 0.07	0.59 ± 0.08	0.53 ± 0.10	0.58 ± 0.07	0.60 ± 0.13
LOF+FreMEn	0.36 ± 0.12	0.39 ± 0.08	0.33 ± 0.12	0.35 ± 0.14	0.40 ± 0.10	0.36 ± 0.13	0.38 ± 0.10
Z-Score+Daily	0.36 ± 0.07	0.45 ± 0.08	0.38 ± 0.07	0.44 ± 0.07	0.43 ± 0.06	0.45 ± 0.07	0.47 ± 0.08
Z-Score+WHyTe	$\mathbf{0.56} \pm 0.20$	$\mathbf{0.77} \pm 0.21$	$\mathbf{0.77} \pm 0.27$	$\mathbf{0.77} \pm 0.27$	$\mathbf{0.87} \pm 0.16$	$\mathbf{0.87} \pm 0.19$	$\mathbf{0.89} \pm 0.06$
Z-Score+Mean	0.17 ± 0.03	0.16 ± 0.01	0.15 ± 0.09	0.16 ± 0.02	0.16 ± 0.01	0.16 ± 0.01	0.16 ± 0.02
Z-Score+Prophet	0.51 ± 0.16	0.55 ± 0.11	0.63 ± 0.08	0.65 ± 0.06	0.64 ± 0.07	0.68 ± 0.05	0.63 ± 0.11
Z-Score+FreMEn	0.38 ± 0.08	0.38 ± 0.06	0.36 ± 0.08	0.34 ± 0.14	0.39 ± 0.07	0.35 ± 0.16	0.39 ± 0.08
Z-Score+Weekly	0.18 ± 0.07	0.22 ± 0.04	0.21 ± 0.06	0.27 ± 0.06	0.34 ± 0.05	0.19 ± 0.06	0.39 ± 0.05

Table 4: Table of Matthews Correlation Coefficients of evaluated detectors on Weekend datasets with different number of measurements in training data.

	60	90	120	150	180	210	240
Prophet Detector	0.35 ± 0.09	0.38 ± 0.09	0.38 ± 0.05	0.40 ± 0.06	0.40 ± 0.04	0.39 ± 0.05	0.39 ± 0.05
FreMEn Detector	0.24 ± 0.04	0.30 ± 0.04	0.35 ± 0.05	0.37 ± 0.05	0.40 ± 0.06	0.43 ± 0.04	0.45 ± 0.06
HyT+LOF	0.04 ± 0.07	0.20 ± 0.13	0.31 ± 0.17	0.26 ± 0.13	0.39 ± 0.09	0.40 ± 0.13	0.47 ± 0.07
HyT+MD	0.16 ± 0.10	0.16 ± 0.10	0.17 ± 0.09	0.20 ± 0.12	0.25 ± 0.07	0.24 ± 0.08	0.25 ± 0.07
HyT+OC-SVM	0.09 ± 0.09	0.23 ± 0.12	0.27 ± 0.09	0.26 ± 0.10	0.37 ± 0.06	0.33 ± 0.08	0.29 ± 0.12
LOF+Daily	0.23 ± 0.06	0.30 ± 0.07	0.27 ± 0.08	0.33 ± 0.07	0.33 ± 0.06	0.33 ± 0.07	0.35 ± 0.09
LOF+WHyTe	0.17 ± 0.19	0.34 ± 0.24	0.35 ± 0.27	0.30 ± 0.24	0.47 ± 0.23	0.41 ± 0.22	0.49 ± 0.18
LOF+Mean	0.05 ± 0.03	0.06 ± 0.04	0.07 ± 0.04	0.04 ± 0.04	0.03 ± 0.04	0.00 ± 0.01	0.00 ± 0.02
LOF+Prophet	$\mathbf{0.42} \pm 0.10$	0.43 ± 0.11	0.43 ± 0.09	0.44 ± 0.09	0.40 ± 0.07	0.37 ± 0.12	0.38 ± 0.10
LOF+FreMEn	0.36 ± 0.07	0.30 ± 0.07	0.34 ± 0.14	0.19 ± 0.21	0.15 ± 0.19	0.07 ± 0.12	0.15 ± 0.18
Z-Score+Daily	0.39 ± 0.05	0.47 ± 0.05	0.44 ± 0.03	0.47 ± 0.05	0.47 ± 0.05	0.50 ± 0.05	0.47 ± 0.04
Z-Score+WHyTe	0.14 ± 0.21	0.31 ± 0.23	0.31 ± 0.29	0.32 ± 0.24	0.48 ± 0.28	$\mathbf{0.53} \pm 0.28$	$\mathbf{0.57} \pm 0.19$
Z-Score+Mean	0.00 ± 0.01	0.01 ± 0.01	-0.01 ± 0.01	0.03 ± 0.01	0.00 ± 0.01	0.01 ± 0.01	0.03 ± 0.01
Z-Score+Prophet	$\mathbf{0.42} \pm 0.13$	$\mathbf{0.49} \pm 0.08$	$\mathbf{0.51} \pm 0.05$	$\mathbf{0.51} \pm 0.06$	$\mathbf{0.50} \pm 0.06$	$\mathbf{0.53} \pm 0.06$	0.54 ± 0.05
Z-Score+FreMEn	0.27 ± 0.03	0.29 ± 0.03	0.34 ± 0.03	0.34 ± 0.04	0.35 ± 0.04	0.35 ± 0.06	0.35 ± 0.05
Z-Score+Weekly	0.01 ± 0.02	0.01 ± 0.02	0.08 ± 0.08	0.03 ± 0.05	0.08 ± 0.06	0.00 ± 0.00	0.15 ± 0.06

Table 5: Table of Matthews Correlation Coefficients of evaluated detectors on Lunch datasets with different number of measurements in training data.

	60	90	120	150	180	210	240
Prophet Detector	0.00 ± 0.03	0.00 ± 0.02	0.01 ± 0.01	0.00 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
FreMEn Detector	$\mathbf{0.37} \pm 0.06$	$\mathbf{0.45} \pm 0.07$	$\mathbf{0.57} \pm 0.07$	$\mathbf{0.60} \pm 0.08$	$\mathbf{0.67} \pm 0.08$	$\mathbf{0.73} \pm 0.07$	$\mathbf{0.74} \pm 0.04$
HyT+LOF	0.01 ± 0.04	0.28 ± 0.24	0.40 ± 0.28	0.53 ± 0.25	0.57 ± 0.25	0.58 ± 0.28	0.65 ± 0.20
HyT+MD	0.18 ± 0.14	0.17 ± 0.16	0.19 ± 0.15	0.14 ± 0.15	0.16 ± 0.16	0.10 ± 0.17	0.09 ± 0.15
HyT+OC-SVM	0.00 ± 0.00	0.16 ± 0.24	0.48 ± 0.29	0.59 ± 0.26	0.55 ± 0.30	0.56 ± 0.25	0.29 ± 0.44
LOF+Daily	0.02 ± 0.07	0.09 ± 0.12	0.12 ± 0.12	0.16 ± 0.08	0.18 ± 0.14	0.18 ± 0.11	0.16 ± 0.13
LOF+WHyTe	0.05 ± 0.14	0.21 ± 0.18	0.21 ± 0.24	0.16 ± 0.22	0.20 ± 0.26	0.06 ± 0.20	0.16 ± 0.21
LOF+Mean	0.02 ± 0.05	0.04 ± 0.11	0.07 ± 0.07	0.01 ± 0.06	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
LOF+Prophet	0.02 ± 0.03	0.00 ± 0.04	0.01 ± 0.04	0.01 ± 0.04	0.01 ± 0.03	0.01 ± 0.06	0.01 ± 0.02
LOF+FreMEn	0.23 ± 0.08	0.09 ± 0.15	0.01 ± 0.02	0.03 ± 0.11	0.01 ± 0.06	0.00 ± 0.02	0.00 ± 0.00
Z-Score+Daily	-0.07 ± 0.06	0.01 ± 0.07	0.01 ± 0.08	0.01 ± 0.06	-0.03 ± 0.05	0.01 ± 0.05	0.02 ± 0.04
Z-Score+WHyTe	0.00 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Z-Score+Mean	0.00 ± 0.05	0.00 ± 0.08	0.00 ± 0.03	-0.06 ± 0.05	0.01 ± 0.06	0.0 ± 0.07	0.01 ± 0.07
Z-Score+Prophet	0.01 ± 0.05	0.01 ± 0.03	0.00 ± 0.04	0.00 ± 0.02	0.00 ± 0.00	0.00 ± 0.01	0.00 ± 0.00
Z-Score+FreMEn	0.23 ± 0.03	0.26 ± 0.06	0.29 ± 0.06	0.30 ± 0.03	0.32 ± 0.03	0.31 ± 0.04	0.33 ± 0.02
Z-Score+Weekly	0.00 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	-0.00 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00

Table 6: Table of Matthews Correlation Coefficients of evaluated detectors on Bimodal datasets with different number of measurements in training data.

6.4 Experiment on Real data

Datasets

- **Training datasets** used for this experiment are described in subsection 4.1. It consists of 7 systematically measured places, each with approximately 150 measurements.
- **Testing datasets** contain 1 day of testing data with a 30 minute period of measurements for each place. Ten percent of collected testing data points were manually changed to a value that did not correspond to reality and were labelled as outliers.

Metrics Matthews Correlation Coefficient is used as a metric to evaluate the anomaly detector's ability to classify data points as outliers.

Process of running the experiment

1. Train all anomaly detectors on all training datasets.
2. Estimate outlier scores in testing data using trained anomaly detection methods.
3. Calculate MCC for each combination of method and testing dataset.
4. Visualize results in a table.

The described process is run for each scenario.

6.4.1 Real data experiment - Results

The table of MCCs for each place and anomaly detector 7 show that no anomaly detector can capture the fundamental nature of outliers in the testing data. We believe this was not caused by the amount of the training data or the quality of anomaly detection models but by the quality of the data itself.

Even though the data quality was insufficient to provide interpretable results on the experiment with the real dataset, some of the models showed a glimpse of their quality. The Prophet Detector, FreMEn Detector and Z-Score+FreMEn showed the most promising overall results of all detectors tested in this experiment. Most of the other anomaly detectors had results comparable to the random detector.

	Albert	Billa	Bistro	Costa	Cukrarna	dm	Dr.Max
Prophet Detector	0.32	**0.48**	**0.32**	-0.09	-0.09	-0.04	-0.04
FreMEn Detector	0.18	0.33	0.23	0.05	0.12	**0.39**	-0.14
HyT+LOF	0.00	0.00	**0.32**	0.12	-0.01	0.18	-0.06
HyT+MD	0.13	0.06	0.16	0.07	0.26	0.06	0.23
HyT+OC-SVM	0.12	0.30	0.30	0.07	0.03	0.34	0.27
LOF+Daily	**0.36**	0.31	0.14	0.28	0.00	0.14	0.30
LOF+HyT	-0.09	0.34	0.27	-0.06	-0.11	-0.04	-0.04
LOF+Mean	0.32	0.46	0.15	0.40	-0.13	0.15	0.18
LOF+Prophet	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LOF+FreMEn	0.00	0.23	0.00	0.00	0.00	0.00	0.00
Z-Score+Daily	0.25	0.33	0.07	-0.02	0.25	0.29	0.18
Z-Score+HyT	0.32	0.23	0.00	-0.06	-0.13	0.00	-0.04
Z-Score+Mean	-0.08	-0.04	0.18	0.05	-0.52	-0.09	0.20
Z-Score+Prophet	0.00	0.00	0.00	-0.08	0.00	0.00	-0.04
Z-Score+FreMEn	0.28	0.25	0.11	**0.33**	**0.41**	0.27	**0.38**
Z-Score+Weekly	0.00	0.00	-0.04	-0.06	0.00	0.00	-0.06

Table 7: Table of Matthews Correlation Coefficients of evaluated detectors on Real dataset for every measured place

7 Conclusion

In this thesis, I researched outlier and novelty detection methods suitable for chronorobotics forecasting methods. Although the chronorobotics proposed spatio-temporal forecasting methods that were successfully applied in autonomous robotics, they do not offer a possibility to uncover novelty or perform outlier detection during preprocessing of data. The lack of a spatio-temporal dataset suitable for anomaly detection led me to perform the experiments mainly over the synthetic datasets. Moreover, the data I used in experiments were time-series without the spatial context. I performed the two-stage experiment. In the first stage, I analysed the behaviour of different combination of forecasting and outlier detection methods with a primary focus to estimate generally usable parameters, namely threshold for the boundary between inliers and outliers. Then, I apply proposed combinations onto the real time-series. It resulted in not very satisfying results.

The best results were obtained by the combination of Prophet with Z-Score and WHyTe with Z-Score. However, only Hypertime Transformation with Local Outlier Factor and FreMEn could detect outliers in a bimodal time-series. As multimodality is expected in the typical spatiotemporal scenarios, it leads me to conclude that WHyTe, with some generalisation of Z-Score, can lead to acceptable and generally usable forecasting with outlier detection.

References

[1] Tomáš Krajník, Tomáš Vintr, George Broughton, Filip Majer, Tomáš Rouček, Jiří Ulrich, Jan Blaha, Veronika Pěčonková, and Martin Rektoris. Chronorobotics: Representing the structure of time for service robots. ISCSIC 2020, New York, NY, USA, 2020. Association for Computing Machinery.

[2] Tomáš Vintr, Sergi Molina, Ransalu Senanayake, George Broughton, Zhi Yan, Jiří Ulrich, Tomasz Piotr Kucner, Chittaranjan Srinivas Swaminathan, Filip Majer, Mária Stachová, et al. Time-varying pedestrian flow models for service robots. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–7. IEEE, 2019.

[3] Gian Diego Tipaldi, Daniel Meyer-Delius, and Wolfram Burgard. Lifelong localization in changing environments. *The International Journal of Robotics Research*, 32(14):1662–1678, 2013.

[4] Tomas Krajnik, Jaime Pulido Fentanes, Grzegorz Cielniak, Christian Dondrup, and Tom Duckett. Spectral analysis for long-term robotic mapping. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3706–3711. IEEE, 2014.

[5] Peer Neubert, Niko Sünderhauf, and Peter Protzel. Appearance change prediction for long-term navigation across seasons. In *2013 European Conference on Mobile Robots*, pages 198–203. IEEE, 2013.

[6] Peter Biber and Tom Duckett. Experimental analysis of sample-based maps for long-term slam. *The International Journal of Robotics Research*, 28(1):20–33, 2009.

[7] Tomáš Krajník, Jaime P Fentanes, Oscar M Mozos, Tom Duckett, Johan Ekekrantz, and Marc Hanheide. Long-term topological localisation for service robots in dynamic environments using spectral maps. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4537–4542. IEEE, 2014.

[8] Nick Hawes, Christopher Burbridge, Ferdian Jovan, Lars Kunze, Bruno Lacerda, Lenka Mudrova, Jay Young, Jeremy Wyatt, Denise Hebesberger, Tobias Kortner, et al. The strands project: Long-term autonomy in everyday environments. *IEEE Robotics & Automation Magazine*, 24(3):146–156, 2017.

[9] Tomáš Vintr, Zhi Yan, Tom Duckett, and Tomáš Krajník. Spatio-temporal representation for long-term anticipation of human presence in service robotics. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2620–2626. IEEE, 2019.

[10] Tomáš Krajník, Jaime P Fentanes, Joao M Santos, and Tom Duckett. Fremen: Frequency map enhancement for long-term mobile robot autonomy in changing environments. *IEEE Transactions on Robotics*, 33(4):964–977, 2017.

[11] Tomáš Krajník, Tomáš Vintr, Sergi Molina, Jaime Pulido Fentanes, Grzegorz Cielniak, Oscar Martinez Mozos, George Broughton, and Tom Duckett. Warped hypertime representations for long-term autonomy of mobile robots. *IEEE Robotics and Automation Letters*, 4(4):3310–3317, 2019.

[12] Leonard Mentzl. Směrové statistiky v predikci kvaziperiodických Časových Řad. Bachelor's thesis, České vysoké učení technické v Praze. Vypočetní a informační centrum., June 2020.

[13] Tomáš Krajnık. Long–term autonomy of mobile robots in changing environments. 2018.

[14] Filip Kubiš. Použití metod modelování časoprostoru v robotice pro predikci poptávky. Bachelor's thesis, June 2020.

[15] Jan Blaha. Odhadování parametrů prediktivních modelů přítomnosti lidí na struktuře prostředí. Bachelor's thesis, June 2020.

[16] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[17] Carl Friedrich Gauss. Theoria motus corporum coelestium. 1809.

[18] Mariette Awad and Rahul Khanna. Support vector regression. In *Efficient learning machines*, pages 67–80. Springer, 2015.

[19] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[20] Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

[21] Tomáš Vintr, Sergi Molina, Ransalu Senanayake, George Broughton, Zhi Yan, Jiří Ulrich, Tomasz Piotr Kucner, Chittaranjan Srinivas Swaminathan, Filip Majer, and Mária Stachová. Time-varying pedestrian flow models for service robots. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–7. IEEE, 2019.

[22] Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

[23] Tomáš Vintr, Kerem Eyisoy, Vanda Vintrová, Zhi Yan, Yassine Ruichek, and Tomáš Krajník. Spatiotemporal models of human activity for robotic patrolling. In *International Conference on Modelling and Simulation for Autonomous Systesm*, pages 54–64. Springer, 2018.

[24] Tomáš Vintr, Zhi Yan, Kerem Eyisoy, Filip Kubiš, Jan Blaha, Jiří Ulrich, Chittaranjan S. Swaminathan, Sergi Molina, Tomasz P. Kucner, and Martin Magnusson. Natural Criteria for Comparison of Pedestrian Flow Forecasting Models. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11197–11204. IEEE, 2020.

[25] Miroslav Kulich, Tomáš Krajník, Libor Přeučil, and Tom Duckett. To explore or to exploit? learning humans' behaviour to maximize interactions with them. In *International Workshop on Modelling and Simulation for Autonomous Systems*, pages 48–63. Springer, 2016.

[26] Tomáš Krajník, Tomáš Vintr, Sergi Molina, Jaime Pulido Fentanes, Grzegorz Cielniak, Oscar Martinez Mozos, George Broughton, and Tom Duckett. Warped hypertime representations for long-term autonomy of mobile robots. *IEEE Robotics and Automation Letters*, 4(4):3310–3317, 2019.

[27] Chris Chatfield et al. Apples, oranges and mean square error. *International Journal of Forecasting*, 4(4):515–518, 1988.

[28] J Scott Armstrong and Fred Collopy. Error measures for generalizing about forecasting methods: Empirical comparisons. *International journal of forecasting*, 8(1):69–80, 1992.

[29] Robert Fildes. The evaluation of extrapolative forecasting methods. *International Journal of Forecasting*, 8(1):81–98, 1992.

[30] Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

[31] Karanjit Singh and Shuchita Upadhyaya. Outlier detection: applications and techniques. *International Journal of Computer Science Issues (IJCSI)*, 9(1):307, 2012.

[32] Richard J Bolton, David J Hand, et al. Unsupervised profiling methods for fraud detection. *Credit scoring and credit control VII*, pages 235–255, 2001.

[33] Davide Azzalini, Alberto Castellini, Matteo Luperto, Alessandro Farinelli, and Francesco Amigoni. Hmms for anomaly detection in autonomous robots. In *International Conference on Autonomous Agents and MultiAgent Systems*, pages 105–113. ACM, 2020.

[34] Sucheta Chauhan and Lovekesh Vig. Anomaly detection in ecg time signals via deep long short-term memory networks. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–7, 2015.

[35] Drausin Wulsin, Justin Blanco, Ram Mani, and Brian Litt. Semi-supervised anomaly detection for eeg waveforms using deep belief nets. In *2010 Ninth International Conference on Machine Learning and Applications*, pages 436–441, 2010.

[36] Ahsan Ijaz and Jongeun Choi. Anomaly detection of electromyographic signals. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(4):770–779, 2018.

[37] Maia Angelova, Chandan Karmakar, Ye Zhu, Sean P. A. Drummond, and Jason Ellis. Automated method for detecting acute insomnia using multi-night actigraphy data. *IEEE Access*, 8:74413–74422, 2020.

[38] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[39] Markus Breunig, Hans-Peter Kriegel, Raymond Ng, and Joerg Sander. LOF: Identifying Density-Based Local Outliers. In *ACM Sigmod Record*, volume 29, pages 93–104, June 2000.

[40] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), July 2009.

[41] Charu C Aggarwal. Outlier analysis. In *Data mining*, pages 237–263. Springer, 2015.

[42] Mingyan Teng. Anomaly detection on time series. In *2010 IEEE International Conference on Progress in Informatics and Computing*, volume 1, pages 603–608. IEEE, 2010.

[43] Charu C Aggarwal. Supervised outlier detection. In *Outlier Analysis*, pages 219–248. Springer, 2017.

[44] Takanori Kudo, Tatsuya Morita, Takahiro Matsuda, and Tetsuya Takine. Pca-based robust anomaly detection using periodic traffic behavior. In *2013 IEEE International Conference on Communications Workshops (ICC)*, pages 1330–1334, 2013.

[45] Bernhard Schölkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, pages 582–588, Cambridge, MA, USA, November 1999. MIT Press.

[46] Chong Zhou and Randy C. Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 665–674, New York, NY, USA, 2017. Association for Computing Machinery.

[47] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. MLSDA'14, page 4–11, New York, NY, USA, 2014. Association for Computing Machinery.

[48] Manish Gupta, Jing Gao, Charu Aggarwal, and Jiawei Han. *Outlier Detection for Temporal Data*. Morgan amp; Claypool Publishers, 2014.

[49] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. Time-series anomaly detection service at microsoft. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Jul 2019.

[50] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Jul 2018.

[51] Alexander E Curtis, Tanya A Smith, Bulat A Ziganshin, and John A Elefteriades. The mystery of the z-score. *AORTA Journal*, 4(4):124, 2016.

[52] John W. Tukey. Variations of box plots. *The American Statistician*, 32(1):12–16, 1978.

[53] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. National Institute of Science of India, 1936.

[54] Peter J. Rousseeuw and Katrien van Driessen. A Fast Algorithm for the Minimum Covariance Determinant Estimator. *Technometrics*, 41(3):212–223, 1999.

[55] Peter Rousseeuw and Bert Zomeren. Unmasking multivariate outliers and leverage points. *Journal of The American Statistical Association - J AMER STATIST ASSN*, 85:633–639, 06 1990.

[56] Vanda Vintrová. *Algoritmy pro vyhledávání lokálně odlehlých pozorování*. PhD thesis, University of Economics in Prague, 2021.

[57] A.L.M. Chiu and Ada Wai chee Fu. Enhancements on local outlier detection. In *Seventh International Database Engineering and Applications Symposium, 2003. Proceedings.*, pages 298–307, 2003.

[58] Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip Gibbons, and Christos Faloutsos. Loci: Fast outlier detection using the local correlation integral. pages 315–326, 01 2003.

[59] Wen Jin, Anthony Tung, Jiawei Han, and Wei Wang. Ranking outliers using symmetric neighborhood relationship. pages 577–593, 04 2006.

[60] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Loop: Local outlier probabilities. pages 1649–1652, 01 2009.

[61] B.W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451, 1975.

[62] David M. W. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation, 2020.

[63] Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[64] Guilherme Campos, Arthur Zimek, Joerg Sander, Ricardo Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30, 07 2016.

[65] Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. Copod: Copula-based outlier detection, 2020.

[66] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. *SIGMOD Rec.*, 29(2):427–438, May 2000.

[67] Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. volume 2431, pages 15–26, 08 2002.

[68] Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2, 2014.

[69] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

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

[71] Stefan Behnel, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn, and Kurt Smith. Cython: The best of both worlds. *Computing in Science & Engineering*, 13(2):31–39, 2011.

[72] Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.