Data Modeling for Outlier Detection

Shahrooz Abghari

Blekinge Institute of Technology Licentiate Dissertation Series
No 2018:04

Data Modeling for Outlier Detection

Shahrooz Abghari

Licentiate Dissertation in
Computer Science

Department of Computer Science and Engineering
Blekinge Institute of Technology
SWEDEN

Department of Computer Science and Engineering
Publisher: Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden
Printed by Exakta Group, Sweden, 2018
ISBN: 978-91-7295-358-1
ISSN:1650-2140
urn:nbn:se:bth-16580

“The more I learn, the more I realize how much I don’t know.”

Albert Einstein

Abstract

This thesis explores the data modeling for outlier detection techniques in three different application domains: *maritime surveillance*, *district heating*, and *online media and sequence datasets*. The proposed models are evaluated and validated under different experimental scenarios, taking into account specific characteristics and setups of the different domains.

Outlier detection has been studied and applied in many domains. Outliers arise due to different reasons such as fraudulent activities, structural defects, health problems, and mechanical issues. The detection of outliers is a challenging task that can reveal system faults, fraud, and save people's lives. Outlier detection techniques are often domain-specific. The main challenge in outlier detection relates to modeling the normal behavior in order to identify abnormalities. The choice of model is important, i.e., an incorrect choice of data model can lead to poor results. This requires a good understanding and interpretation of the data, the constraints, and the requirements of the problem domain. Outlier detection is largely an unsupervised problem due to unavailability of labeled data and the fact that labeled data is expensive.

We have studied and applied a combination of both machine learning and data mining techniques to build data-driven and domain-oriented outlier detection models. We have shown the importance of data preprocessing as well as feature selection in building suitable methods for data modeling. We have taken advantage of both supervised and unsupervised techniques to create hybrid methods. For example, we have proposed a rule-based outlier detection system based on *open data* for the maritime surveillance domain. Furthermore, we have combined cluster analysis and regression to identify manual changes in the heating systems at the building level. Sequential pattern mining for identifying contextual and collective outliers in online media data have also been exploited. In addition, we have proposed a minimum spanning tree clustering technique for detection of groups of outliers in online media and sequence data. The proposed models have been shown to be capable of explaining the underlying properties of the detected outliers. This can facilitate domain experts in narrowing down the scope of analysis and understanding the reasons of such anomalous behaviors. We have also investigated the reproducibility of the proposed models in similar application domains.

Preface

Included Papers

This thesis consists of four papers. In **Paper I**, the author has been one of the main drivers of the paper while in papers **II-IV** he has been the main driver. The studies in all papers have been developed and designed under the guidance of the supervisors and domain experts. The formatting of the published papers included in this thesis have been changed to achieve a consistent style in the thesis.

Paper I Kazemi, S., Abghari, S., Lavesson, N., Johnson, H., & Ryman, P. "Open data for anomaly detection in maritime surveillance". *Expert Systems with Applications*, (40)14 (2013), pp. 5719-5729.

Paper II Abghari, S., Garcia-Martin, E., Johansson, C., Lavesson, N., & Grahn, H. "Trend analysis to automatically identify heat program changes". *Energy Procedia*, 116 (2017), pp. 407-415. Also published as Paper V.

Paper III Abghari, S., Boeva, V., Lavesson, N., Grahn, H., Gustafsson, J., & Shaikh, J. "Outlier detection for video session data using sequential pattern mining". In *Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining: Workshop On Outlier Detection De-constructed*, 2018, London, UK.

Paper IV Abghari, S., Boeva, V., Lavesson, N., Grahn, H., Ickin, S., & Gustafsson, J. "A minimum spanning tree clustering approach for outlier detection in event sequences". In *The 17th IEEE International Conference on Machine Learning and Applications: Special Session on Machine Learning Algorithms, Systems and Applications*, December, 2018, Orlando, Florida, USA. (Accepted for publication.)

Other research contributions that are related to this thesis but not included:

Paper V Abghari, S., Garcia Martin, E., Johansson, C., Lavesson, N., & Grahn, H. Trend analysis to automatically identify heat program changes". In *The 15th International Symposium on District Heating and Cooling*, 2016, Seoul, Korea.

Poster I Abghari, S., Boeva, V., Lavesson, N., Grahn, H., Gustafsson, J., & Shaikh,J. "Anomaly detection in video session data". In *The Fifth Swedish Workshop on Data Science*, 2017, Gothenburg, Sweden.

Acknowledgements

I would like to thank the people who have supported me to make this thesis possible. First and foremost, I would like to thank my supervisors Professor Niklas Lavesson, Professor Håkan Grahn, and Professor Veselka Boeva for their trust, patience, guidance, and valuable feedback. I appreciate the opportunities you provided for me to learn and grow both professionally and personally. Thanks to all my friends and colleagues who have supported me through having discussions and commenting on my work all these years. In particular, I would like to say thank you to Eva García-Martín (Ebba) for her positive energy, supportive attitude and being someone that I can trust. Johan Silvander, thank you for giving me hope and positive energies during times that were tough.

I would like to say thanks to the Swedish Knowledge Foundation for funding my research within the project "Scalable resource-efficient systems for big data analytics" under grant 20140032. Furthermore, I would like to extend my gratitude to Jörgen Gustafsson, the research manager at Ericsson Research Stockholm, and his team for their time, support and guidance. I also appreciate Christian Johansson, the CTO of NODA Intelligent Systems AB, for providing resources and the opportunity for research collaboration.

Last but not least, I would like to say thanks to my family who have always been supportive. I would like to thank my loving girlfriend Amber for her patience, understanding me and always being there.

Karlskrona, Sweden

October 9, 2018

Contents

1

Introduction

Outlier detection has been studied and applied in different fields to detect anomalous behaviors. An outlier is a data point which is significantly different from other surrounding points [1–3]. Outliers can happen due to different reasons such as human error, fraudulent behavior, mechanical faults, and instrument error. Regardless of the source of the outliers, their detection can reveal system faults, fraud, and interesting patterns in the data. The detected outliers can assist experts in narrowing down the scope of analysis and understanding the root cause(s) of such anomalous behaviors.

Almost all outlier detection techniques create a model representing the normal patterns in the data[1] to be able to detect whether a given data point is an outlier or not. There are several factors such as the nature of the input data, availability of the labeled data together with the constraints, and the requirements of the outlier detection problem at hand that make the data modeling challenging [1].

The nature of the input data affects the choice of data model. Input is a collection of data instances together with their types or attributes. There are different types of attributes such as categorical (which includes *nominal* and *ordinal*), numerical (which includes *discrete* and *continuous*), and binary. Each data instance may consist of a single attribute (univariate) or multiple attributes (multivariate) [1].

Outliers can be divided into three groups, namely *point outliers*, *contextual outliers*, and *collective outliers*. The nature of the desired outlier requires a specific outlier detection technique to be identified. Outlier detec-

[1] There are some studies that model abnormalities of the data [4–8]. Some authors referred to this technique as novelty detection. It can be considered as a semi-supervised task since the algorithm is taught the normal class, however, it learns to recognize abnormality [1, 2].

tion is largely an unsupervised problem, i.e., examples of outliers are not available to learn the best model. This makes the process of data modeling difficult compared to supervised scenarios where labeled data are available. Moreover, the way outliers are reported, e.g., as some sort of outlying scores or labels has an important role in data modeling. Evidently, the choice of a data model is important. It is often data-specific which requires a good understanding and interpretation of the data [1, 3].

This thesis explores the data modeling of the outlier detection problem in three different application domains. Each of the studied domains has unique constraints and requirements which demands validation with different experimental setups and scenarios. Outlier detection techniques are domain-dependent. They usually are developed for certain application domain. We also explore in this thesis the reproducibility of the proposed methods in similar domains. Initially, we *investigate the importance of data modeling for outlier detection techniques in surveillance, trend analysis and fault detection.* In addition, *the reproducibility of the proposed approaches in similar domains is investigated.*

As discussed above, the outlier detection problem is data-specific. In this thesis, two different sources of data are used. The majority of data used in our experiments is provided by companies involved in the conducted studies. Such data is closed and not publicly available, *closed data*. The other source, on the other hand, relates to *open data*, data that can be freely accessed, used, re-used, and shared by anyone for any purpose. This thesis, specifically *investigates the application of open data as a complimentary source for maritime surveillance domain.*

The main contribution of this thesis is its focus on studying different outlier detection scenarios and developing data-driven and domain-aware outlier detection models. The thesis contains four papers. In **Paper I**, we develop a rule-based model for detecting anomaly in the maritime surveillance domain. In addition, we study the application of open data as a complementary resource in identifying anomalous activities. In **Paper II**, we propose a hybrid approach for detecting manual changes and trend analysis in the district heating domain. In **Paper III** and **Paper IV**, we study sequence data and sequences of events that might cause issues at the system level in online media data and smart meter data.

2 Background

2.1 Machine Learning

The term *machine learning* (ML) is first introduced by Arthur L. Samuel [9] in his famous paper "Some Studies in Machine Learning Using the Game of Checkers" in 1959. Samuel defines ML as the *"field of study that gives computers the ability to learn without being explicitly programmed."* A more precise definition is proposed by Tom Mitchell in his book *Machine Learning* in 1997 [10]: "*A computer program is said to* ***learn*** *from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.*" In this definition T refers to the class of tasks, P the measure of performance to be improved, and E the source of experience, i.e., the training data. Peter Flach uses the following general definition in his book [11]: "*Machine learning is the systematic study of algorithms and systems that improve their knowledge or performance with experience.*" The process of learning begins with data and information in the form of observations and real-world interactions in order to grasp patterns in data and make better decisions in the future based on the observed data. As all these definitions suggest, in ML the primary goal is to allow the computer programs to learn automatically without human intervention and to adjust actions accordingly while improving their learning over time.

Machine learning is a branch of artificial intelligence (AI). It was born from *pattern recognition* and *computational learning theory* in artificial intelligence. ML is closely related to *computational statistics* and *mathematical optimization*. In the 1990s, the field of machine learning shifts from knowledge-driven approaches further toward data-driven approaches. Computer programs that can learn and methods such as support vector machines [12], random forest [13], and long short-term memory recurrent

neural networks [14] become popular.

Machine learning methods can be divided into two main categories based on the availability of the labeled training data to a learning algorithm as follows:

Supervised learning algorithms are trained in the presence of labeled examples, such as input where the target class is known. In supervised learning, the goal is to learn a general rule on how to map inputs to outputs. Typical supervised learning tasks are classification and regression. Supervised learning can be divided further into three sub-categories when labeled data is partially available to the learning algorithm or the learning process is restricted to special feedback. **Semi-supervised learning** is used for the same applications as supervised learning. However, both labeled data (in a small portion) and unlabeled data are used for training. The main reasons relate to the fact that providing labeled training data is time consuming and requires experts' knowledge which makes the process of data labeling expensive. An **active learning** algorithm has the ability to select its own training examples from the provided dataset and asks from some oracle or labeling source to label them. In **Reinforcement learning**, an algorithm discovers through trial and error which actions can maximize the expected rewards over a given period of time [11].

Unsupervised learning is used on data that has no labels. In this scenario the learning algorithm on its own explores the input data in order to find some structures such as discovering hidden patterns [11]. Clustering analysis is an unsupervised learning task that is applied in this thesis and is explained in Section 2.4.1.

2.2 Data Mining

Data mining refers to the discovery of *models* for data [15] and extracting knowledge from large amount of data [16]. A *model* can be one of several types. The following are the important directions in modeling.

- **Statistical Modeling.** The term *data mining* was first used by statisticians. Originally, data mining has a *negative* meaning and refers to extracting information that was not supported by the data.

Today, statisticians apply data mining to construct statistical models to further study the underlying distribution of data [15].

- **Machine Learning.** Data mining is often used interchangeably with machine learning. Machine learning tries to understand the structure of the data. Data mining, instead, applies different machine learning algorithms to identify previously unknown patterns and extract insights from data [15].

- **Computational Approaches to Modeling.** Such modeling is looking at data mining as an algorithmic problem, i.e., the model of the data can simply explain complex queries about it. *Summarization* and *feature extraction* are two examples of such modeling. In summarization the aim is to provide a clear and approximate summary of the data [15]. Regarding feature extraction, we can refer to *frequent itemsets mining* that is explained in Section 2.4.5.

2.3 Outlier Detection

According to Grubbs [17], an outlier is an observation that deviates significantly from other members of the sample in which it occurs. A similar definition provided by Barnett and Lewis [18], stating that an outlier observation is the one which appears to be inconsistent with the reminder of that set of data. Hawkins [19] defined an outlier as a distinct observation that is seemed to be generated by a different mechanism. The detection of outliers requires an expert's knowledge to model the normal behaviors or patterns in data.

The problem of finding patterns that are distinct from the majority of the data is called outlier detection. These distinct patterns are often referred to as outliers or anomalies[1]. Outlier detection is related but distinct of noise detection. Noise can appear as *attribute noise* (implicit errors and missing values introduced by the measurement tools), *class noise* (mislabeled instances), or a combination of both. Errors and exceptions that occur during data collection and data preprocessing phases represent noise which should be repaired or eliminated from the data [20]. Outliers, on the other hand, can be

[1] Outliers are also referred to as aberrations, contaminants, deviations, discordant observations, error, exceptions, noise, novelty, peculiarities, or surprises in different domains [1, 2].

considered as interesting and/or unknown patterns hidden in data which may lead to new insights, the discovery of system faults, or fraudulent activities. While some works do not separate noise and outliers [2], others refer to noise as weak outliers, whereas strong outliers considered as anomalies that are more inserting for analysts [3].

Figure 2.1: The spectrum of data from normal data to outliers (Adopted from [3].)

Outliers can be classified into three categories [1, 16, 21]:

- *Point Outliers*: An individual data point that is distinct from the entirety of the dataset can be considered as a point (global) outlier. Point outliers are the simplest types of outliers.

- *Contextual Outliers*: A data point that deviates significantly with respect to a specific context or condition is referred to as contextual or conditional outlier. In contextual outlier, the data instances are evaluated by considering two groups of attributes 1) *Contextual attributes*, the context or neighborhood for the instance, 2) *Behavioral attributes*, the characteristics of the instance.

- *Collective Outliers*: A collection (sequence) of related data points that deviates significantly from the entire dataset. In a collective outlier, the individual data points in the sequence may or may not be outliers by themselves.

Outlier detection methods have been suggested and designed for many application domains such as network intrusion detection, fraud detection in identification of criminal activities related to credit card, mobile phone, and insurance claims, medical condition monitoring, fault detection, satellite image analysis, sensor network, surveillance and traffic monitoring and other data mining tasks.

Outlier detection techniques can be classified into three groups based on the availability of the labeled data [1, 2]: **1)** In the absence of prior knowledge of the data, unsupervised learning methods are used to determine outliers. The initial assumption is that normal data represents a significant portion of the data and is distinguishable from faults or error; **2)** In the presence of labeled data, both normality and abnormality are modeled. This approach refers to supervised learning; **3)** Define what is normal and only model normality. This approach is known as semi-supervised learning since the algorithm is trained by labeled normal data, however, it is able to detect outliers or abnormalities. Semi-supervised outlier detection techniques are more widely used compared to supervised techniques due to an imbalanced number of normal and abnormal labeled data.

The output of an outlier detection is one of the following two types [1–3]:

- *Scores.* Scoring outlier detection techniques provide a score quantifying the degree to which each data point is considered as an outlier. Such techniques provide a ranked list of outliers which facilitate domain experts to analyze the top few outliers or define a domain-specific threshold to select the outliers.
- *Labels.* This category of techniques assign a binary label to each data point indicating whether it is an outlier or not.

2.4 Methods

In this thesis, we use different machine learning and data mining techniques, distance measures, and performance measures to perform the outlier detection task. In this section, we discuss these methods and measures.

2.4.1 Cluster Analysis

Cluster analysis is the task of grouping data without prior knowledge into coherent groups, i.e., the degree of similarity between instances in each group are high while the similarity between different groups should be reduced [11]. Traditional clustering algorithms can be grouped into 5 categories, namely partitioning, hierarchical, model-based, density-based, and grid-based methods. Due to the limitation of the traditional clustering algorithms in handling massive amounts of data in applications such as data stream

mining, incremental clustering methods have been proposed. These clustering methods are capable of analyzing data instances one at a time and assigning them to the existing clusters. Detailed surveys regarding clustering methods can be found in [22–24]. In this thesis, we used affinity propagation and k-means algorithms to perform the clustering task. Both algorithms belong to the partitioning algorithms category.

Affinity Propagation

The affinity propagation (AP) algorithm [25] works based on the concept of exchanging messages between data points until a good set of exemplars (the most representative of a cluster) and corresponding clustering solution appears. The exchanged messages at each step assist AP to choose the best samples as exemplars and to assign suitable data points to each exemplar. AP adapts the number of clusters based on the data. However, the number of clusters can be controlled by the *preference* parameter. That is, a high value of the preference will cause AP to form many clusters, while a low value will lead to a small number of clusters. If the preference value is not provided the median similarity or the minimum similarity will be used. Unlike most clustering algorithms such as k-means [26] and k-medoids [27] that require the number of clusters as an input, AP is able to estimate the number of clusters based on the data provided. AP can create clusters of different shapes and sizes. Similar to k-medoids, affinity propagation finds exemplars (the selected data points) that are the representative of the clusters [28].

k-means

k-means clustering algorithm, also known as Lloyd's algorithm [26], partitions n data points into k clusters. The algorithm proceeds by randomly choosing a set of k centroids. The set is iteratively revised until the within-cluster sum-of-squares errors are minimized. k-means requires k to be known. In **Paper III**, we use the *Silhouette Index* to determine the optimal number of clusters in k-means. Silhouette Index measure is explained in Section 5.5.

2.4.2 Rule-based models

Rule-based models belong to the logical machine learning models. Rule learning can be performed in two ways. The first approach is similar to decision tree learning, which relates to finding a combination of literals that

can cover a homogeneous set of examples and assigning a label to each rule. The second approach starts with the class we want to learn and find rule bodies that can cover the examples of that class. The first approach will lead to a model consisting of an ordered rule list while in the second approach the model is a collection of unordered rule sets [11]. In **Paper I**, we used a collection of validated expert rules to detect anomalous behavior in the maritime surveillance domain. The rule set is defined from series of scenarios that can occur for passenger and cargo vessels.

2.4.3 Regression

In a *classification* task, the aim is to learn an approximation $\hat{c} : X \to L$, where $L = \{C_1, C_2, ..., C_k\}$ is a discrete set of *class labels* similar to the true labeling learned from the training data labels. In *regression*, the target variable space is the set of real numbers. The aim in a regression task is to learn a function estimator, *regressor*, $\hat{f} : X \to \mathbb{R}$ from the training instances [11].

Support Vector Regression

The support vector machine (SVM) algorithm is based on statistical learning theory. SVM is a state-of-the-art algorithm, which belongs to a group of supervised learning methods that can solve different ML tasks such as classification, pattern recognition and regression [12]. An extended version of SVM for regression tasks is called support vector regression (SVR). SVR uses the training data to find the regression line that best fits the data. Using an epsilon-intensive loss function, SVR produces a decision boundary, a subset of training data which is called support vectors (SVs), in order to determine a tube with radius ε fitted to the data. In other words, epsilon defines how well the regression line fits the data by ignoring errors as long as they are less than ε. In **Paper II**, SVR is used to learn the heat demand in each building using the outdoor temperature and the secondary supply temperature.

2.4.4 Minimum Spanning Tree

In a complete weighted graph where the distance between edges are the traversing weight, a minimum spanning tree is a sub-set of edges that connect all the vertices together without any cycles that has the minimum total

edge weight. Prime's algorithm [29] and Kruskal's algorithm [30] are two examples of greedy algorithms for identifying a minimum spanning tree (MST). Minimum spanning trees have direct applications in network design such as computer networks and telecommunications networks. It is also used for cluster analysis [31–33]. In **Paper IV**, we use MST to identify groups of outliers by removing the longest edge(s) of a tree.

2.4.5 Frequent Itemset Mining

The application of frequent itemset mining for market-basket analysis was first introduced by Agrawal et al. [34] in 1993. The aim of such analysis is to reveal the customers' shopping habits and to find out which sets of products are frequently bought together. The frequent itemset mining can be formulated as follows: let $\mathcal{I} = \{i_1, i_2, ..., i_n\}$ be a set of all items and $\mathcal{T} = \{t_1, t_2, ..., t_j, ..., t_m\}$ a transaction database, where t_j is a set of items that has been bought by a customer ($t_j \subseteq \mathcal{I}$). The aim is to find those sets of items that occur frequently in most of the shopping baskets considering s, the user-specified *support threshold*.

The *support* for a *k-itemset* X, which consists of k items from $\mathcal{I}$, is the number of transactions that contain X as a subset, i.e., $_{ST}(X) = |\{t_j | X \subseteq t_j \wedge t_j \in \mathcal{T}\}|$. Note that the support of X can also be defined as the *relative support* which is the ratio of the number of transactions containing X to the total number of transactions in the database $\mathcal{T}$, i.e., $_{RelST}(X) = \frac{_{ST}(X)}{|\mathcal{T}|}$, such X is frequent if and only if its support is equal or greater than s.

2.4.6 Sequential Pattern Mining

Originally in frequent itemset mining, the order of items in the itemsets is unimportant. Looking at the market-basket analysis, the goal is to find frequent sets of items that are bought together. However, there are some situations in which the order of items inside the itemset is important such as sequence databases. A sequence database consists of ordered sequences of items listed with or without a concrete notion of time [35]. Sequential pattern mining, the problem of finding interesting frequent ordered patterns, was first introduced in 1995 [36].

Let $\mathcal{I} = \{i_1, i_2, ..., i_n\}$ be a set of all items. A sequence α defined as $\langle a_1, a_2, ..., a_j, ..., a_m \rangle$, where a_j is an itemset. Each itemset a_j represents a

set of items that happened at the same time. A sequence $\alpha = \langle a_1, a_2, ..., a_m \rangle$ is a subsequence of $\beta = \langle b_1, b_2, ..., b_n \rangle$ if and only if there exist integers $1 \leq i_1 < i_2 < ... < i_n \leq m$ and $a_1 \subseteq b_{i_1}, a_2 \subseteq b_{i_2}, ..., a_m \subseteq b_{i_m}$ [36]. Given a sequence database $\mathcal{T} = \{s_1, s_2, ..., s_n\}$, the support for α is the number of sequences in $\mathcal{T}$ that contain α as a subsequence. Consequently, α is a frequent sequential pattern if its support is equal or greater than the user-specified support threshold.

Mining frequent patterns in a large database can lead to generating a huge number of patterns that satisfy the user-specified support threshold. This is due to the fact that if a pattern is frequent, its sub-patterns are also frequent. To mitigate this problem, *closed* and *maximal* frequent pattern mining has been proposed [35]. A frequent pattern α is called:

1. a closed frequent pattern in the database $\mathcal{T}$ if and only if none of its super-patterns have the same support as α,

2. a maximal frequent pattern in the database $\mathcal{T}$ if and only if none of its super-patterns is frequent [37], [35].

2.4.7 Sequential Pattern Mining Algorithms

Since the introduction of frequent itemset mining and the Apriori algorithm [34], several extensions of this algorithm were developed for both frequent itemset mining and sequential pattern mining. In general, there are two main categories of algorithms suitable for frequent pattern mining: 1) *Apriori-based algorithms* and 2) *Pattern-growth algorithms*. Additionally, from a frequent pattern mining point of view, a sequence database can represent the data either in a *horizontal data format* or *vertical data format* [38]. Therefore, based on these two data formats Apriori-based algorithms can expand to *horizontal data format algorithms* such as AprioriAll [36], and GSP [39] and *vertical data format algorithms* such as SPADE [40], and SPAM [41]. Apriori-based algorithms generate large sets of candidates and repeatedly scan the database for mining sequential patterns which require a lot of memory [42]. To solve this problem, pattern-growth approach as an extension of the FP-growth algorithm [42] for frequent itemset mining without candidate generation was proposed. Pattern-growth algorithms such as FreeSpan [43], and PrefixSpan [44] work in a divide-and-conquer fashion

and repeatedly divide the database into a set of smaller *projected databases* and mine them recursively.

We use PrefixSpan for pattern mining in **Paper III** and **Paper IV**. PrefixSpan applies a *prefix-projection* method to find sequential patterns. Given a sequence database $\mathcal{T}$ and a user-specified threshold, the database is first scanned in order to identify all frequent items in sequences. All of these frequent items are considered as length-1 sequential pattern. After that, the search space is divided into a number of subsets based on the extracted prefixes. At last, for each subset a corresponding *projected database* is created and mined recursively.

2.4.8 Distance Measures

In this thesis, we use different distance measures to calculate the similarity between strings. In **Paper I**, we use *JaroWinkler* to match vessels' information from different data sources.

JaroWinkler [45] is a variation of the *Jaro* metric [46, 47] that is commonly used for name matching in record-linkage [48]. Jaro calculates the number of common characters, c, within the half-length of the longer string and the number of transpositions, t, for two strings s_1 and s_2 as:

$$Jaro(s_1, s_2) = \frac{1}{3}\left(\frac{c}{|s_1|} + \frac{c}{|s_2|} + \frac{c-t}{c}\right) \tag{2.1}$$

The JaroWinkler improves the Jaro metric by assigning extra weight to the common prefix at the beginning of two strings to be compared.

$$JaroWinkler(s_1, s_2) = Jaro(s_1, s_2) + lp(1 - Jaro(s_1, s_2)), \tag{2.2}$$

where l is the length of common prefix at the beginning of the two strings and p is the prefix assigned weight. The standard value of p is equal to 0.1.

In **Paper III**, we compare the application of *Fast Dynamic Time Warping* (FastDTW) and *Levenshtein Distance* (LD) for constructing a similarity matrix and performing cluster analysis. In **Paper IV**, LD is used for cluster analysis.

FastDTW [49] is an approximation of DTW [50] that has a linear time and space complexity. It detects an accurate optimal alignment between two time series and finds the corresponding regions between them. FastDTW reduces the resolution of the time series repeatedly with averaging adjacent pairs of points. Then it takes a minimum-distance warp path at a lower resolution and projects to a higher resolution. The projected warp path is refined and repeatedly projected onto incrementally higher resolutions until a full warp path is found. FastDTW is calculated as:

$$D(i,j) = Dist(i,j) + min \begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases} \tag{2.3}$$

The *Dist* is the Euclidean distance between two points. In this thesis, due to availability of categorical data $Dist(i,j)$ represents equality, i.e., if $i = j$ *Dist* is equal to 0, otherwise 1.

Levenshtein Distance [51] is defined to be the minimum number of edit operations (insertions, deletions, or substitutions) required to transform one string into another. The LD of two strings s and t is

$$LD(i,j) = min \begin{cases} LD(i-1,j) + cost_{deletion} \\ LD(i,j-1) + cost_{insertion} \\ LD(i-1,j-1) + cost_{substitution} \end{cases} \tag{2.4}$$

Each operation has unit cost, except the substitution when $s_i = t_j$ that has zero cost.

3

Data Sources: Open vs. Closed

According to *Open Definition*[1] the term *open* means anyone can freely use, re-use, modify and redistribute for any reasons. That is, *open data* is data that can be freely accessed, used, re-used, and shared by anyone for any purpose. More specifically, open data requires to be 1) legally open and available under an open license that permits anyone to access, use and share it, 2) technically open and usable through no more than the cost of reproduction and in machine-readable format.

According to the *Open Data Institute* (ODI), data is closed if it can only be accessed by its subject, owner or holder. This includes data that is only available to an individual, or a group of people or within an organization for specific reasons. *Closed data* should not be shared for security reasons, or since it contains personal information.

There is a third category that is referred to as *shared data*. According to ODI, shared data is available to:

a. named people or organizations (Named access),

b. specific groups who meet certain criteria (Group-based access),

c. anyone under terms and conditions that are not *open* (Public access).

Figure 3.1 shows that the data spectrum provided by ODI ranges from *closed* to *shared* to *open* data. Many individuals and organizations generate and collect different types of data in a daily basis to perform their tasks. Governments, however, in this regard have a particular importance mainly due to the quantity and centrality of the data they produce and also because

[1] `https://www.opendefinition.org/`

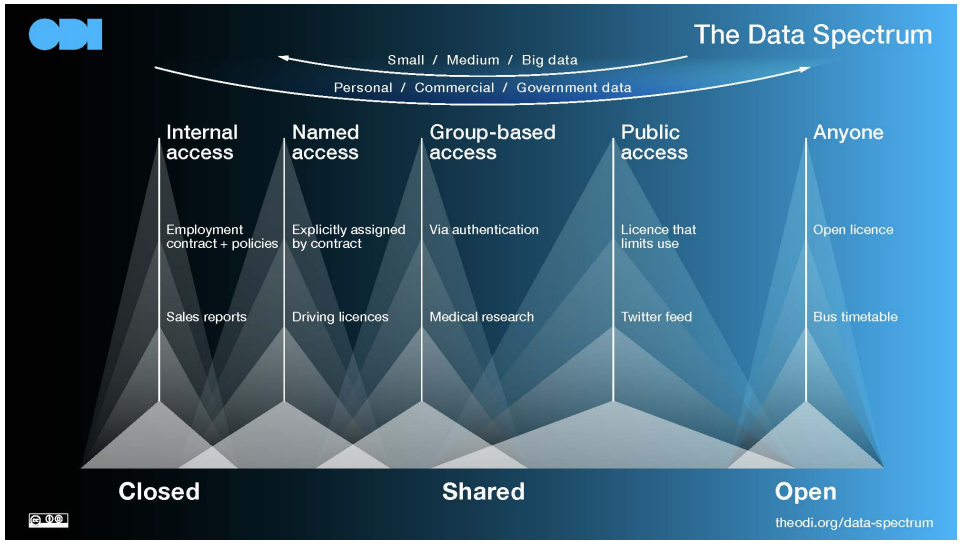


Figure 3.1: The ODI data spectrum (Reproduced from ODI.)

most of this data is public data by law. Through making this data open and available for everyone, many individuals and organizations, especially governmental organizations, can benefit. Such data can improve efficiency and effectiveness of government services which also reduces the costs. It brings transparency and can lead to new private products and services.

In **Paper I**, the application of open data in the maritime surveillance domain is studied. The majority of the studies that are done in the context of outlier detection in this domain used sensor data and mainly the automatic identification system (AIS) data to find anomalies in coastal regions. Detection of some suspicious activities such as smuggling requires vessel traffic data beyond the coastal region. Maritime authorities in each country have overall information of maritime activities in their surveillance area. But exchanging information among different countries is a complicated procedure because of the diverse regulation of data protection in each land. Therefore, using data sources that are free from legislative procedures can be a good solution for providing information that belongs to the regions outside the land territory. Furthermore, all the information about maritime activities is not recorded in the authorities' databases or reported to them. There are numerous open data sources, on the other hand, consisting of different websites, blogs, and social networks that can be useful for observing the hidden aspects of maritime activities.

4

Related Work

Outlier detection techniques have been studied and successfully applied in different domains. There exists a considerable number of studies that provide a comprehensive, and structured overview of the state-of-the-art methods and applications for outlier detection [1, 2, 52, 53]. In this thesis, we focused on three different domains, namely maritime surveillance (MS), district heating (DH), and online media and sequence datasets. Outlier detection techniques are domain-specific, i.e., applying a technique developed in one domain to another domain is not an easy task. This relates mainly to the nature of the data, availability of the labeled data, and type of outliers that should be detected. These factors are determined by the application domain and their identifications by domain experts can lead to choose a suitable data modeling.

4.1 Maritime Surveillance Domain

Maritime surveillance is the effective understanding of all maritime activities that could impact the security, safety, economy or environment[1]. Maritime transport handles over 80% of the volume of global trade[2]. Along with the development of the maritime transport system, the threats to maritime security such as illegal fishing and pollution, terrorism, smuggling activities and illegal immigration are increasing correspondingly.

In recent years, the number of studies that address the use of outlier detection in the MS domain is increasingly growing. The outlier detection

[1] Integrating Maritime Surveillance, common information sharing environment, `https://www.ec.europa.eu/maritimeaffairs/policy/integrated_maritime_surveillance/documents/integrating_maritime_surveillance_en.pdf`

[2] United Nations Conference on Trade and Development (UNCTAD), Review of Maritime Transport 2011, `https://www.unctad.org/en/Docs/rmt2011_en.pdf`

techniques can be divided into two groups, namely *data-driven* and *knowledge-driven* approaches. There are a couple of works that propose knowledge-based outlier detection systems with different representation techniques and reasoning paradigms such as rule-based, description logic and case-based reasoning [54–56]. A prototype for a rule-based expert system based on the maritime domain ontologies was developed by Edlund et al. [57]. The proposed prototype could detect some of the anomalies regarding the spatial and kinematic relation between objects such as simple scenarios for hijacking, piloting and smuggling. Another rule-based prototype was developed by Defence R&D Canada [58, 59]. The aforementioned prototype employed various maritime situational facts about both the kinematic and static data in the domain to make a rule-based automated reasoning engine for finding anomalies. One of the popular data-driven outlier detection approaches is the Bayesian network [60–62]. Johansson and Falkman [61] used the kinematic data for creating the network; however, in the work that was done by Fooladvandi et al. [60] expert's knowledge as well as the kinematic data were used in the detection process. Moreover, Lane et al. [62] presented the detection approaches for five unusual vessel behaviors and the estimation of the overall threat was performed by using a Bayesian network. Unsupervised learning techniques have been widely used for data-driven outlier detection such as trajectory clustering [63], self organizing map [64] and fuzzy ARTMAP neural network [65]. Some statistical approaches, such as Gaussian mixture model [66], hidden Markov model [67], adaptive kernel density estimator [68] and precise/imprecise state-based outlier detection [63] have been used in this context.

There is a number of studies that employed data fusion techniques to fuse data from different sensors in outlier detection systems [69–72]. In these studies, the surveillance area was restricted to the coastal regions and the combination of data from AIS, synthetic aperture radar, infra-red sensors, video and other types of radar was used in the fusion process to obtain the vessel tracks. Furthermore, there are some other works that focused on the fusion of both sensor and non-sensor data, e.g., expert's knowledge [60, 73–77]. Riveiro and Falkman [77] introduced a normal model of vessel behavior based on AIS data by using a self organizing map and a Gaussian mixture model. According to the model, the expert's knowledge about the common characteristic of the maritime traffic was captured as if – then rules and the outlier detection procedure was supposed to find the deviation from

the expected value in the data. Lefebvre and Helleur [75] used radar data with user's knowledge about the vessels of interests. The sensor data were modelled as track and the non-sensor data were modelled as templates. The track-template association was done by defining mathematical models for tracks and using fuzzy membership functions for association possibilities. Mano [76] proposed a prototype for the maritime surveillance system that could collect data from different types of sensors and databases and regroup them for each vessel. Sensors like AIS, high frequency surface wave radar and classical radars and databases such as environmental database, Lloyd's Insurance and TF2000 Vessel database were included in this prototype. By using multi-agent technology an agent was assigned to each vessel and anomalies could be detected by employing a rule-based inference engine. When the combination of outliers exceeded a threshold, vessel status was reported to the user as an anomaly. The work presented by Ding et al. [74], proposed an architecture of a centralized integrated maritime surveillance system for the Canadian coasts. Sensors and databases included in this architecture were: high frequency surface wave radar, automatic dependant surveillance reports, visual reports, information sources, microwave radar, and radar sat. A common data structure was defined for storing data that were collected from different sensors. Andler et al. [73], also described a conceptual maritime surveillance system that integrated all available information such as databases and sensor systems (AIS, long-range identification and tracking, intelligence reports, registers/databases of vessels, harbours, and crews) to help users to detect and visualize anomalies in the vessel traffic data on a worldwide scale. Furthermore, the authors suggested using open data in addition to other resources in the fusion process.

4.2 Districts Heating Domain

A district heating (DH) system is a centralized system with the aim of producing space heating and hot tap water for consumers based on their demand within a limited geographic area. A DH system consists of three main parts: production units, distribution network, and consumers. The heated water supplied in a production unit circulates through the distribution network and will be available to consumers. The main aim of a DH system is to minimize the cost and pollution by considering consumers' demand and producing just the necessary amount of heat. Hence, being able to

predict the heat demand can assist production units to plan better. However, modeling the heat demand forecasting is a challenging task, since water does not move fast. In some situations, the distribution of heated water can take several hours. Moreover, there are a number of factors that affect the forecast accuracy and need to be considered before any plan for production units can be constructed. Some of these factors include [78], [79]:

1. Weather condition, mainly the outdoor temperature
2. Social behavior of the consumers
3. Irregular days such as holidays
4. Periodic changes in conditions of heat demand such as seasonal, weekly and day-night

Fumo [80] pointed out in his review two commonly used techniques for energy demand estimation, namely; forward (classical) and data-driven (inverse) techniques. The first approach describes the behavior of systems by applying mathematical equations and known inputs to predict the outputs. In contrast, data-driven techniques use ML methods to learn the system's behavior by building a model with training data in order to make predictions.

Dotzauer [79] introduced a very simple model for forecasting heat demand based on outdoor temperature and social behavior. He showed that the predictions of his simple model were comparable with complicated models such as autoregressive moving average model (ARMA). The author concluded that better predictions can be achieved by improving the weather forecasts instead of developing complicated heat demand forecasting models.

In general, different ML methods and techniques have been used to predict the heat demand. Some of the most popular prediction models are autoregressive moving average (ARMA) [81], support vector regression (SVR) [82], [83], multiple linear regression (MLR) [84] and artificial neural network (ANN) [85], [86]. In [83] the authors compared four supervised ML methods for building short-term forecasting models. The models are used to predict heat demand for multi-family apartment buildings with different horizon values between 1 to 24 hours ahead. The authors concluded that SVR achieves the best performance followed by MLR in comparison to feed forward neural networks (FFNN), and regression trees methods. Recently, Provatas

et al. [87], proposed the usage of on-line ML algorithms in combination with decision tree-based ML algorithms for heat load forecasting in a DH system. The authors investigated the impact of two different approaches for heat load aggregation. The results of the study showed that the proposed algorithm has a good prediction result. In another study [88], the authors showed the application of a context vector (CV) based approach for forecasting energy consumption of single family houses. The proposed method is compared with linear regression, K-nearest neighbors (KNN) and SVR methods. The results of the experiment showed that CV performed better in most cases followed by KNN and SVR. The authors concluded the proposed solution can help DH companies to improve their schedule and reduce operational costs.

There are a number of studies that focus on the application of decision support (DS) systems in domains such as DH and mainly related to advanced energy management [89], [90], [91], [92], [93], [94]. In these studies, the main focus is on forecasting and optimization methods that facilitate and support the decision-making processes to increase the energy management quality and bring considerable savings. Furthermore, there are some other works that focused on DH network design [95], [96]. Bordin et al. [95] presented a mathematical model to support DH system network planning by selecting an optimal set of new users to be connected to a thermal network that maximizes revenues and minimizes infrastructure and operational costs.

4.3 Online Media Domain and Sequence Datasets

The Internet has transformed almost every aspect of human society by enabling a wide range of applications and services such as online video streaming. Subscribers of such services spend a substantial amount of time online to watch movies and TV shows. This has required online video service providers (OVSPs) to continuously improve their services and equipment to satisfy subscribers' high expectation. According to a study performed by Krishnan and Sitaraman [97], a 2-second delay in starting an online video program causes the viewers to start abandoning the video. For each extra second delay beyond that, the viewers' drop-off rate will be increased by 5.8%. Thus, in order for OVSPs to address subscribers' needs it is important to monitor, detect, and resolve any issues or anomalies that can significantly affect the viewers when watching requested video programs. Analyzing

massive amounts of video sessions for identifying such abnormal behaviors is like finding a needle in a haystack.

Barbará et al. [98] proposed an intrusion detection system that applies a frequent itemset technique to discover sets of items that are available in most data chunks. Using a clustering algorithm, these items that are considered as attack-free traffic, are divided into different groups based on their similarities. After creating the clusters, an outlier detection technique is applied to all the data points checking each instance against the set of clusters. Instances that do not belong to any clusters are presumed to be attacks. Recently, Rossi et al. [99] proposed an anomaly detection system for the smart grid domain similar to the one considered in [98]. The method proposed by Rossi et al. uses frequent itemset mining on different event types collected from smart meters to separate normal and potential anomalous data points. For further evaluation, a clustering technique with Silhouette Index analysis is applied to detect anomalies.

Hoque et al. [100] developed an anomaly detection system for monitoring daily in-home activities of elderly people called *Holmes*. The proposed system learns a resident's normal behavior by considering variability of daily activities based on their occurrence time (e.g., day, weekdays, weekends) and applying a context-aware hierarchical clustering algorithm. Moreover, *Holmes* learns temporal relationships between multiple activities with the help of both sequential pattern mining and itemset mining algorithms. New scenarios can be added based on resident and expert's feedback to increase the accuracy of the system.

There are several clustering algorithms capable of detecting noise and eliminating it from the clustering solution such as DBSCAN [101], CRUE [102], ROCK [103], and SNN [104]. Even though such techniques can be used to detect outliers, the main aim for the clustering algorithm is to perform the partitioning task rather than identifying outliers.

This led to proposing clustering-based techniques that are capable of detecting: **1)** single-point outliers such as the application of Self Organizing Maps for clustering normal samples and identifying anomalous samples [105], and Expectation Maximization [106] for identifying the performance problems in distributed systems or **2)** groups of outliers such as the intrusion detection proposed by [107].

The application of MST has been studied by researchers in different fields including cluster analysis and outlier detection [33, 108–111]. A two-phase clustering algorithm is introduced for detecting outliers by Jiang et al. [108]. In the first phase, a modified version of the k-means algorithm is used for partitioning the data. The modified k-means creates $k + i$ clusters, i.e., if a new data point is far enough from all clusters (k, number of clusters defined by the user), it will be considered as a new cluster (the $(k + i)^{th}$ cluster where, $i > 0$). In the second phase, an MST is built where, the tree's nodes represent the center of each cluster and edges show the distance between nodes. In order to detect outliers, the longest edge of the tree is removed. The sub-trees with a few number of clusters and/or smaller clusters are selected as outliers.

Wang et al. [109] developed an outlier detection by modifying k-means for constructing a spanning tree similar to an MST. The longest edges of the tree are removed to form the clusters. The small clusters are regarded as potential ouliters and ranked by calculating a density-based outlying factor.

A spatio-temporal outlier detection for early detection of critical events such as flood through monitoring a set of meteorological stations is introduced in [111]. Using geographical information of the data, a Delaunay triangulation network of the stations is created. The following step limits the connection of nodes to their closest neighbors while preventing far nodes from being linked directly. In the next step, an MST is constructed out of the created graph. In the final step, the outliers are detected by applying two statistical methods to detect exactly one or multiple outliers.

5 Research Methodology

5.1 Research Problem

This thesis focuses on studying and developing of data models for the outlier detection problems. Such problems are often domain-specific. Understanding and modeling of the available data is the core of any outlier detection system. The main challenge in the outlier detection problems relates to the definition of normal behavior. In general, defining a normal region that represents every possible normal behaviors is very difficult [1]. Moreover, due to unavailability of the labeled data for training and validation, outlier detection is often categorized as an unsupervised problem.

5.2 Research Questions

The main questions addressed by this thesis are: 1) *How can data understanding and modeling facilitate the detection of outliers with respect to finding outlier explanatory components?*, 2) *Can open data be used for data modeling in outlier detection techniques?* The following two main questions are broken down into more specific research questions that are investigated in different papers.

RQ1. *How can **open data** complement **closed data** for anomaly detection in the maritime surveillance domain?*

The reason for studying this question is that maritime authorities in each country only have overall information of maritime activities in their surveillance area. Exchanging information, between different countries is often a complicated procedure because of the diverse regulation of data protection in each land. Therefore, using data sources that are free from legislative procedures can be a good solution

for providing information that belongs to the regions outside the land territory. In **Paper I**, we study the application of *open data* as a complementary source for outlier detection in the maritime domain.

RQ2. *What is the positive and negative impacts of the open data sources?(that is, what is the increase in true negatives and positives in comparison to the increase in false negatives and positives?)*

This question investigates the generalizability of the system results in the real world by considering 1) *accuracy* as the degree to which the proposed outlier detection system is able to distinguish between the normal and anomalous activities and 2) *validity* as the degree to which the system results are true in real world. The following research question is addressed in **Paper I**.

RQ3. *How can contextual and collective outliers be used to identify manual changes in heating systems?*

Detection of *point outliers* in the presence of noise can generate a lot of false alarms. This led us to this question, *how can collective outliers be identified in our case study?* Moreover, since in Sweden the meteorological definition of seasons are more suitable compared to calendrical definition, we focused on contextual outliers. In **Paper II**, we address this question by studying the district heating domain.

RQ4. *Can sequential pattern mining be used to determine the underlying properties of the detected outliers?*

This research question concerns both **Paper III** and **Paper IV**. The papers study the application of sequential pattern mining in identifying outliers in the absence of labeled data. Moreover, it is shown how sequential pattern mining can facilitate the domain experts in understanding the underlying properties of detected outliers.

RQ5. *What are the benefits of identifying outliers as groups rather than analyzing individual data points?*

Based on the study conducted in **Paper III**, we have realized that the proposed approach is not able to detect outliers that occur in more than one segment. This approach first, divides the data into equal-sized segments and then extracts frequent sequential patterns from each segment. The initial assumption here, is that those patterns

that occur in more than one segments are normal and low frequent patterns are the potential outliers. In order to mitigate this issue, analyzing and extracting frequent sequential patterns from different length of segments, e.g., daily, two-day, and weekly segments are required. Moreover, calculating an outlier score for each of suspicious data points increases the run-time of the proposed approach. Therefore, in **Paper IV**, we study group outlier detection techniques.

5.3 Research Method

In this thesis, we have designed and performed a set of experiments to study the performance of the proposed outlier detection models. An *experiment* is a test or series of tests with some *factors* to adjust that affect the output. Factors divide into *controllable factors* such as the choice of the learning algorithm and *uncontrollable factors* that add undesired effects to the process such as the noise in the data [112]. Figure 5.1 shows the construction of an outlier detection technique.

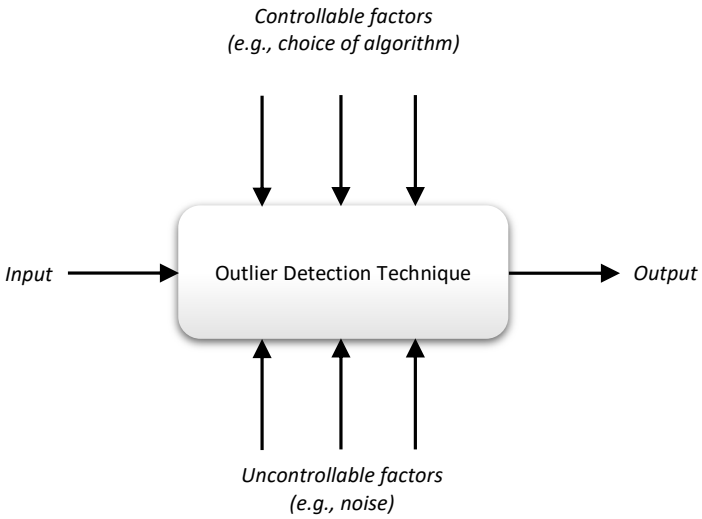


Figure 5.1: Constructing an outlier detection technique (Adopted from [112].)

In each study, we have initially formulated our research questions and have defined the study objectives. Next, we have designed experimental

setups in coordination with the domain experts. After that, data collection, preprocessing and feature selection have been done. The latter has been performed under the guidance of the domain experts to choose relevant features in model construction. We have analyzed and evaluated the preprocessed data to choose suitable machine learning and data mining techniques. In this thesis, parameters of the selected algorithms often have been adjusted through running them with different values. In some studies, we have run the selected algorithms with their default parameters such as the application of SVR in **Paper II**. Moreover, in **Paper III**, Silhouette Index has been used for finding the optimal number of clusters for the k-means algorithm. Finally, we have built a model, evaluated the results, and validated our findings with the domain experts to make sure the proposed model perform correctly. Figure 5.2 summarizes the experimental design applied in this thesis.

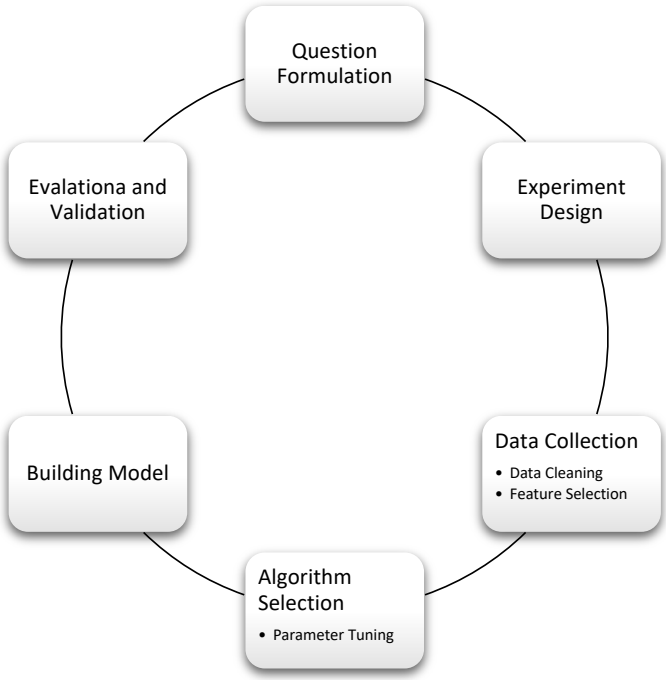


Figure 5.2: Experimental design for this thesis

5.4 Datasets

In **Paper I**, we used the following data sources: *Marinetraffic*[1] for having access to semi-real time information based on AIS systems. The pilotage schedule related to Stockholm area are collected from *Swedish Maritime Administration (Sjöfartsverket)*[2]. *Ports of Stockholm*, *Kapellskär*, and *Nynäshamn*, information regarding vessels in port and expected arrival[3]. *Port of Norrköping*, information regarding vessels in port[4] and expected arriving vessels[5]. *Port of Helsinki*, information regarding cargo vessels in port and expected cargo vessels' arrival[6], expected passenger vessels' arrival and departure[7], and international cruise vessels' arrival, departure and those that have visited the port before[8]. *Port of Tallinn*, information regarding vessels in port[9], expected cargo vessels' arrival[10], expected passenger vessels' arrival[11], and expected passenger vessels' departure[12].

The data used in **Paper II** consists of daily average measurements from 9 buildings equipped with the *NODA*[13] controller. The buildings are located in the south of Sweden. These measurements include, electricity consumption, outdoor temperature and secondary supply temperature.

In **Paper III** and **Paper IV**, video session data provided by *Ericsson AB* is used. The data is stored as a transaction dataset. Each row of the dataset contains of session ID, video ID, date and time of an occurring video event together with its type. After preprocessing of the data, the dataset is transformed into a sequence dataset, i.e., each row represents a video session with its starting and ending time, session duration, video programs that is chosen, and a sequence of all event types that happened during the session.

[1] http://www.marinetraffic.com/ais/
[2] https://eservices.sjofartsverket.se/lotsinfopublic/lotsning_frames.asp
[3] http://www.portsofstockholm.com/vessel-calls/
[4] http://www.norrkopingshamn.se/en/ankomstinformation/fartyg-i-hamnen
[5] http://www.norrkopingshamn.se/en/ankomstinformation/fartyg-i-hamnen-anlop
[6] https://www.portofhelsinki.fi/en/cargo-traffic-and-ships/arrivals-and-departures-cargo
[7] https://www.portofhelsinki.fi/en/passengers/arrivals-and-departures
[8] https://www.portofhelsinki.fi/en/passengers/international-cruise-ships
[9] http://www.ts.ee/?op=ships_in_port&lang=eng
[10] http://www.ts.ee/?op=cargo_ships_arrivals&lang=eng
[11] http://www.ts.ee/?op=passenger_ship_arrivals&lang
[12] http://www.ts.ee/?op=passenger_ship_departures&lang=eng
[13] https://www.noda.se/

Apart from video session data, in **Paper IV**, we used a publicly available dataset containing smart meter data provided by *Elektro Ljubljana*, a power distribution company in Slovenia [113]. The data is stored as a transaction dataset. Data contains device ID, event data(short explanation regarding each event), event ID, and event timestamps. The dataset is transformed into a sequence dataset, i.e., each row represents a daily activity of a device, time of recoding the first and last event types, and a sequence of all event types that happened during its daily activity.

5.5 Evaluation Measures

In this thesis, different evaluation measures are used for analysis and assessment of the results. In **Paper I** and **Paper II**, the accuracy of the system is calculated as the degree to which the measurements of a quantity correctly describe the exact value of that quantity. In other words, accuracy is the proportion of truly labeled results among the total number of cases examined. Accuracy is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{5.1}$$

where true positive (TP) and true negative (TN) refer to correctly labeled cases. A false positive (FP) happens when a real negative case is misclassified as positive. A false negative (FN) occurs when a real positive case is labeled as negative [114].

In **Paper II**, mean absolute error is used as a performance measure to evaluate the accuracy of support vector regression in terms of predicting the secondary supply temperature.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|actual_i - predicted_i| \tag{5.2}$$

In this equation the *actual* refers to the measured secondary supply temperature by the controller system, *predicted* refers to the estimated secondary supply temperature by the proposed DS system, and n is the total number of predicted instances.

In **Paper III**, *Silhouette Index (SI)* [115] is applied as an internal validation measure due to unavailability of the ground truth labels. SI

evaluates the tightness and separation of each cluster and measures how well an object fits the available clustering. For each i, let $a(i)$ be the average dissimilarity of i to all other objects in the same cluster. Let us now consider $d(i, C)$ as an average dissimilarity of i to all objects of a cluster C. After computing $d(i, C)$ for all clusters, the one with the smallest average dissimilarity is denoted as $b(i)$. Such cluster also refer to *neighboring cluster* of i. The Silhouette Index score of i, $s(i)$, is obtained by combing $a(i)$ and $b(i)$ as follows:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{5.3}$$

The $s(i)$ has values in a range of $[-1, 1]$. A score close to one implies that the object is well clustered. When $s(i)$ is about zero, this indicates the object is on the decision boundary between two neighboring clusters. The worst situation occurs when $s(i)$ is close to -1. This indicates that the object is misclassified and assigned to the wrong cluster.

The average $s(i)$ for all objects i belonging to the same cluster shows how tightly those objects are grouped. The average $s(i)$ for all objects i in the whole dataset judges the quality of the generated clustering solution.

5.6 Validity Threats

In this thesis, there are some issues that can threat the validity of the results and they should be considered before applying the proposed techniques in any systems and performing the evaluation and validation. *Construct validity* refers to the extent to which the results of a study reflect the theory or the concept behind [116]. The main issue that may threat the construct validity is the design and reliability of the implementation. The results can be affected by the potential faults that may happen in the implementation either because of programming faults or lack of parameter tuning. In addition, the inaccurate nature of *open data* that are used in **Paper I** may have some effect on the results. The open data can have errors due to mistakes of the human operator. They do not follow a similar format and can be unavailable for a while or are not updated immediately. The undesirable effect of the open data can be reduced through applying a common data representation format, preprocessing of the data, and using approximate string matching techniques.

For decreasing the effect of programming faults, the validity of the implemented techniques are tested different times with both real and manipulated data that contain anomalies. There are also some parameters in different studies that are needed to be selected correctly. In **Paper I**, in order to match the vessel names, we use a string matching technique and a similarity threshold to determine whether the two names are identical. In papers **III** and **IV**, we can refer to the *user-specified support threshold* for extracting frequent sequential patterns from the data. Choosing an inappropriate value for such parameters will lead to incorrect results.

The other validity threat, which can occur while performing the evaluation and validation, targets both the internal and external validities. *Internal validity* ensures that the observed relationship between the treatment and the outcome is due to a causal relationship and is not because of an uncontrolled factor. *External validity* refers to the ability of generalizing the results of the study to other domains, times or places [116]. The threat to internal and consequently the external validity may occur if the data that are used in the evaluation process are biased and are not representative of the population. In such situation generalizing the results of the treatments to the whole system is unrealistic. To prevent this issue a proper sampling technique ensuring a realistic representation of the studied population in both the experiment and the validation is required.

6

Results

Paper I investigates the potential of open data sources as a complementary resource for outlier detection in the maritime surveillance domain. A framework for outlier detection is proposed based on the usage of open data sources along with other traditional sources of data. The proposed framework can be generalized to similar application domains due to its modularized and general design. Based on the proposed outlier detection framework and the algorithms for implementing the expert rules, an outlier detection system is developed. The validity of the results is investigated by the subject matter experts from the Swedish coastguard. The validation results show that the majority of the system's evaluated anomalies (64.47%) are true alarms. Moreover, a potential information gap in the closed data sources is observed during the validation process. That is, for 9.21% of the evaluated anomalies there are no corresponding data in the authorized databases by the coastguard. Despite the high number of true alarms, the number of false alarms is also considerable which is mainly because of the inaccurate open data (26.32%). Since the main purpose of the study was to focus the analysis and investigation on two defined scenarios, it is not possible to draw any conclusions about the generalizability of the results.

Paper II studies trend analysis in the district heating domain using a hybrid method. A decision support system is proposed to identify manual changes. The system applies k-means to detect the operational status of the heating system (on or off) by partitioning the consumed energy at each building. The support vector regression is used to identify manual changes with respect to the operational status of the heating system. To achieve this goal and to avoid generating false alarms in presence of noise, changes are monitored for some days. Hence, in this study only those deviations that last for at least 3 consecutive days are marked as manual changes. In this study, we focus on both contextual and collective outliers. That is, a manual

change is identified with respect to the operational status of a heating system (context) and a set of at least three consecutive measurements that do not conform with the heat signature of a building (collective outlier).

In **Paper III** and **Paper IV**, we propose two approaches for identifying outliers in sequence datasets. In **Paper III**, the proposed approach first segments the data and extracts frequent sequential patterns in each segment. The extracted patterns are divided into two groups by considering the fact whether they occur in more than one segment. The initial assumption is that those patterns that occur in more than one segment are normal. These patterns are used for modeling the normal behavior. At the end, low frequent sequential patterns are matched into the clustering model. The goodness-of-fit of each pattern is evaluated using Silhouette Index. Those patterns with scores close to the lowest possible value assumed to be outliers.

In **Paper IV**, we focus on identifying groups of outliers rather than individually evaluating the outlierness of each low frequent sequential pattern. That is, after extracting frequent sequential patterns and partitioning them, a minimum spanning tree is applied on top of the clustering solutions. By removing the longest edges of the tree, sub-trees with fewer nodes and smaller clusters in size are identified as outliers. Group outlier detection can reduce the time complexity of the system. In this paper, we study and evaluate our model on two different sequence datasets: smart meter data and video session data. In that way, we also show the reproducibility of the proposed model in two different application domains.

In addition, the obtained results on video session data are validated by domain experts. The validation results show that 12 out of 18 sessions are correctly labeled by the proposed approach. Further analysis of the experts' comments reveals that assessing the quality of a video session is not a trivial task and sometimes can be subjective. The validation also shows that the proposed approach is able to identify video sessions that are significantly different from the majority of the sessions due to occurrence of some specific event types. The identification of such sequences of events can assist the domain experts to understand underlying properties of the detected outliers.

7

Conclusions and Future Work

This thesis contains four research papers with the focus on data modeling for outlier detection. The outlier detection techniques have been designed, developed, and evaluated in three different application domains: *maritime surveillance*, *district heating*, and *online media and sequence data*.

The main contribution of this thesis is on understanding and modeling of data for the outlier detection problem. We have studied and applied a combination of both machine learning and data mining techniques to build data-driven and domain-oriented outlier detection models. We have shown the importance of data preprocessing as well as feature selection in developing suitable methods for data modeling. In this thesis, both supervised and unsupervised techniques have been applied to create hybrid methods. The proposed methods have been demonstrated to be capable of explaining the underlying properties of the detected outliers.

The main contribution in **Paper I** is threefold: i) the use of *open data* for outlier detection in the maritime surveillance domain, ii) the design of a framework for outlier detection based on integration of open and closed data sources, and iii) the development of a rule-based outlier detection system based on the proposed framework. In **Paper II**, we evaluate an approach to detect manual changes in heating systems at the building level. Our main contribution here is the application of cluster analysis in combination with support vector regression in the district heating domain. In **Paper III**, we exploit sequential pattern mining for identifying unexpected patterns in online streaming media data. Our main contribution in this paper is the combination of sequential pattern mining, cluster analysis and Silhouette Index measure for unsupervised contextual and collective outlier detection in sequence data. In **Paper IV**, we propose a minimum spanning tree clustering approach in combination with sequential pattern mining for

identifying groups of outliers in online streaming media and sequence data. Additionally, the reproducibility of the proposed model in similar application domains is investigated.

In this thesis, **Paper I** and **Paper II** study outlier detection as a supervised learning problem. In **Paper I**, the implemented system uses expert rules to detect anomalous behavior in the maritime surveillance domain. The proposed framework in this study suggests the usage of both knowledge-driven approaches to detect previously seen anomalies and data-driven approaches to detect unseen anomalies. In **Paper II**, we apply cluster analysis and a regression method to learn the expected heat demand at each building to detect manual changes in the heating system. Outdoor temperature and energy consumption of the buildings are used for modeling the heat demand.

Papers **III** and **IV**, apply unsupervised learning techniques as the core for data modeling. In both papers, *sequential pattern mining* and *clustering analysis* is used for extracting and grouping frequent sequential patterns. In **Paper III**, outliers are detected through matching suspicious patterns with the clustering solutions and evaluating goodness-of-fit using Silhouette Index. In **Paper IV**, on the other hand, the focus is on detecting groups of outliers using minimum spanning tree on top of the clustering solutions and cutting the longest edges of the tree. That is, sub-trees with fewer clusters and smaller sized clusters can be recognized as outliers due to their significant uniqueness and distance from the majority of the data.

A possible future direction can be the application of *kernel methods* for pattern analysis and outlier detection. Real world problems often require a nonlinear methods to be solved. The kernel-based learning methods embed data into a feature space where the nonlinear patterns can be discovered as linear relations. The efficiency, robustness, and statistical stability of the Kernel-based algorithms can be integrated with machine learning and data mining techniques for building powerful analysis tools. Another future direction can be the focus on interpretation and understanding of the detected outliers. That is, given an outlier, it is important to determine *how* it differs from the remainder of the data. Finding a proper outlier explanatory components can facilitate domain experts in verifying outliers and understanding the underlying of detected outliers. One solution can be using an attribute subset scoring to determine possible explanations for the

detected outliers [117].

References

[1] V. Chandola, A. Banerjee, and V. Kumar. “Anomaly detection: A survey”. In: *ACM Computing Surveys* 41.3 (2009), p. 15.

[2] V. Hodge and J. Austin. “A survey of outlier detection methodologies”. In: *Artificial Intelligence Review* 22.2 (2004), pp. 85–126.

[3] C. C. Aggarwal. “Outlier analysis”. In: *Data mining*. Springer. 2015, pp. 237–263.

[4] N. Japkowicz, C. Myers, M. Gluck, et al. “A novelty detection approach to classification”. In: *IJCAI*. Vol. 1. 1995, pp. 518–523.

[5] T. Fawcett and F. Provost. “Activity monitoring: Noticing interesting changes in behavior”. In: *Proc. of the fifth ACM SIGKDD Int’l Conf. on Knowledge Discovery and Data Mining*. ACM. 1999, pp. 53–62.

[6] C. Warrender, S. Forrest, and B. Pearlmutter. “Detecting intrusions using system calls: Alternative data models”. In: *Proc. of the 1999 IEEE Symp. on Security and Privacy*. IEEE. 1999, pp. 133–145.

[7] D. Dasgupta and N. S. Majumdar. “Anomaly detection in multidimensional data using negative selection algorithm”. In: *Proc. of the 2002 Congress on Evolutionary Computation*. Vol. 2. IEEE. 2002, pp. 1039–1044.

[8] D. Dasgupta and F. Nino. “A comparison of negative and positive selection algorithms in novel pattern detection”. In: *Int’l Conf. on Systems, Man, and Cybernetics*. Vol. 1. IEEE. 2000, pp. 125–130.

[9] A. L. Samuel. “Some studies in machine learning using the game of checkers”. In: *IBM J. of Research and Development* 3.3 (1959), pp. 210–229.

[10] T. M. Mitchell et al. *Machine learning. 1997*. McGraw-Hill Education, 1997.

[11] P. Flach. *Machine learning: The art and science of algorithms that make sense of data*. Cambridge University Press, 2012.

[12] V. Vapnik. *Statistical learning theory. 1998*. Wiley, New York, 1998.

[13] T. K. Ho. “Random decision forests”. In: *Proc. of the third Int’l Conf. on Document Analysis and Recognition*. Vol. 1. IEEE. 1995, pp. 278–282.

[14] S. Hochreiter and J. Schmidhuber. “Long short-term memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[15] J. Leskovec, A. Rajaraman, and J. D. Ullman. *Mining of massive datasets*. Cambridge University Press, 2014.

[16] J. Han, J. Pei, and M. Kamber. *Data mining: Concepts and techniques*. Elsevier, 2011.

[17] F. E. Grubbs. “Procedures for detecting outlying observations in samples”. In: *Technometrics* 11.1 (1969), pp. 1–21.

[18] V. Barnett and T. Lewis. *Outliers in statistical data*. Wiley, 1974.

[19] D. M. Hawkins. *Identification of outliers*. Vol. 11. Springer, 1980.

[20] B. Sluban. “Ensemble-based noise and outlier detection”. PhD Dissertation. Jožef Stefan International Postgraduate School, 2014. URL: `http://slais.ijs.si/theses/2014-03-19-Sluban.pdf`.

[21] X. Song, M. Wu, C. Jermaine, and S. Ranka. “Conditional anomaly detection”. In: *IEEE Transactions on Knowledge and Data Engineering* 19.5 (2007), pp. 631–645.

[22] R. Xu and D. Wunsch. “Survey of clustering algorithms”. In: *IEEE Transactions on Neural Networks* 16.3 (2005), pp. 645–678.

[23] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras. “A survey of clustering algorithms for big data: Taxonomy and empirical analysis”. In: *IEEE Transactions on Emerging Topics in Computing* 2.3 (2014), pp. 267–279.

[24] M. Angelova. “Clustering techniques for analysis of large datasets”. In: *Fundamental Sciences and Applications* (2017), p. 113.

[25] B. J. Frey and D. Dueck. “Clustering by passing messages between data points”. In: *Science* 315.5814 (2007), pp. 972–976.

[26] S. Lloyd. “Least squares quantization in PCM”. In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137.

[27] L. Kaufman and P. Rousseeuw. *Clustering by Means of Medoids*. 1987.

[28] U. Bodenhofer, A. Kothmeier, and S. Hochreiter. “APCluster: An R package for affinity propagation clustering”. In: *Bioinformatics* 27.17 (2011), pp. 2463–2464.

[29] R. C. Prim. “Shortest connection networks and some generalizations”. In: *Bell System Technical* 36.6 (1957), pp. 1389–1401.

[30] J. B. Kruskal. “On the shortest spanning subtree of a graph and the traveling salesman problem”. In: *Proc. of the American Mathematical Society* 7.1 (1956), pp. 48–50.

[31] T. Asano, B. Bhattacharya, M. Keil, and F. Yao. “Clustering algorithms based on minimum and maximum spanning trees”. In: *Proc. of the Fourth Annual Symp. on Computational Geometry*. ACM. 1988, pp. 252–257.

[32] G. W. Flake, R. E. Tarjan, and K. Tsioutsiouliklis. “Graph clustering and minimum cut trees”. In: *Internet Mathematics* 1.4 (2004), pp. 385–408.

[33] G.-W. Wang, C.-X. Zhang, and J. Zhuang. “Clustering with Prim’s sequential representation of minimum spanning tree”. In: *Applied Mathematics and Computation* 247 (2014), pp. 521–534.

[34] R. Agrawal, T. Imieliński, and A. Swami. “Mining association rules between sets of items in large databases”. In: *Acm sigmod record*. Vol. 22. 2. 1993, pp. 207–216.

[35] J. Han, H. Cheng, D. Xin, and X. Yan. “Frequent pattern mining: Current status and future directions”. In: *Data Mining and Knowledge Discovery* 15.1 (2007), pp. 55–86.

[36] R. Agrawal and R. Srikant. “Mining sequential patterns”. In: *Proc. of the 11th Int’l Conf. on Data Engineering*. IEEE. 1995, pp. 3–14.

[37] C. Borgelt. “Frequent item set mining”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2.6 (2012), pp. 437–456.

[38] W. Shen, J. Wang, and J. Han. “Sequential pattern mining”. In: *Frequent Pattern Mining*. Springer, 2014, pp. 261–282.

[39] R. Srikant and R. Agrawal. “Mining sequential patterns: Generalizations and performance improvements”. In: *Advances in Database Technology–EDBT’96* (1996), pp. 1–17.

[40] M. J. Zaki. “SPADE: An efficient algorithm for mining frequent sequences”. In: *Machine Learning* 42.1 (2001), pp. 31–60.

[41] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. “Sequential pattern mining using a bitmap representation”. In: *Proc. of the Eighth ACM SIGKDD Int’l Conf. on Knowledge Discovery and Data Mining*. 2002, pp. 429–435.

[42] J. Han, J. Pei, and Y. Yin. “Mining frequent patterns without candidate generation”. In: *ACM sigmod record*. Vol. 29. 2. 2000, pp. 1–12.

[43] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.-C. Hsu. “FreeSpan: Frequent pattern-projected sequential pattern mining”. In: *Proc. of the Sixth ACM SIGKDD Int’l Conf. on Knowledge Discovery and Data Mining*. 2000, pp. 355–359.

[44] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. “Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth”. In: *Proc. of the 17th Int’l Conf. on Data Engineering*. 2001, pp. 215–224.

[45] W. E. Winkler. “The state of record linkage and current research problems”. In: *Statistical Research Division, US Census Bureau*. Citeseer. 1999.

[46] M. A. Jaro. “Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida”. In: *J. of the American Statistical Association* 84.406 (1989), pp. 414–420.

[47] M. A. Jaro. “Probabilistic linkage of large public health data files”. In: *Statistics in Medicine* 14.5-7 (1995), pp. 491–498.

[48] W. E. Winkler. “Overview of record linkage and current research directions”. In: *Bureau of the Census*. Citeseer. 2006.

[49] S. Salvador and P. Chan. “Toward accurate dynamic time warping in linear time and space”. In: *Intelligent Data Analysis* 11.5 (2007), pp. 561–580.

[50] H. Sakoe and S. Chiba. “Dynamic programming algorithm optimization for spoken word recognition”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26.1 (1978), pp. 43–49.

[51] V. I. Levenshtein. “Binary codes capable of correcting deletions, insertions, and reversals”. In: *Soviet physics doklady*. Vol. 10. 8. 1966, pp. 707–710.

[52] Y. Zhang, N. Meratnia, and P. Havinga. “Outlier detection techniques for wireless sensor networks: A survey”. In: *IEEE Communications Surveys & Tutorials* 12.2 (2010), pp. 159–170.

[53] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han. “Outlier detection for temporal data: A survey”. In: *IEEE Transactions on Knowledge and Data Engineering* 26.9 (2014), pp. 2250–2267.

[54] A. B. Guyard and J. Roy. “Towards case-based reasoning for maritime anomaly detection: A positioning paper”. In: *Proc. of The IASTED Int’l Conf. on Intelligent Systems and Control*. Vol. 665. 006. 2009, p. 1.

[55] M. Nilsson, J. Van Laere, T. Ziemke, and J. Edlund. “Extracting rules from expert operators to support situation awareness in maritime surveillance”. In: *The 11th Int’l Conf. on Information Fusion*. IEEE. 2008, pp. 1–8.

[56] J. Roy and M. Davenport. “Exploitation of maritime domain ontologies for anomaly detection and threat analysis”. In: *Int’l Conf. on Waterside Security*. IEEE. 2010, pp. 1–8.

[57] J. Edlund, M. Grönkvist, A. Lingvall, and E. Sviestins. “Rule-based situation assessment for sea surveillance”. In: *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*. Vol. 6242. Int’l Society for Optics and Photonics. 2006, p. 624203.

[58] J. Roy. “Anomaly detection in the maritime domain”. In: *Optics and Photonics in Global Homeland Security IV*. Vol. 6945. Int’l Society for Optics and Photonics. 2008, 69450W.

[59] J. Roy. “Rule-based expert system for maritime anomaly detection”. In: *Sensors, and Command, Control, Communications, and Intelligence Technologies for Homeland Security and Homeland Defense IX*. Vol. 7666. Int’l Society for Optics and Photonics. 2010, 76662N.

[60] F. Fooladvandi, C. Brax, P. Gustavsson, and M. Fredin. "Signature-based activity detection based on Bayesian networks acquired from expert knowledge". In: *The 12th Int'l Conf. on Information Fusion*. IEEE. 2009, pp. 436–443.

[61] F. Johansson and G. Falkman. "Detection of vessel anomalies-a bayesian network approach". In: *The Third Int'l Conf. on Intelligent Sensors, Sensor Networks and Information*. IEEE. 2007, pp. 395–400.

[62] R. O. Lane, D. A. Nevell, S. D. Hayward, and T. W. Beaney. "Maritime anomaly detection and threat assessment". In: *The 13th Conf. on Information Fusion*. IEEE. 2010, pp. 1–8.

[63] A. Dahlbom and L. Niklasson. "Trajectory clustering for coastal surveillance". In: *The Tenth Int'l Conf. on Information Fusion*. IEEE. 2007, pp. 1–8.

[64] M. Riveiro, G. Falkman, and T. Ziemke. "Improving maritime anomaly detection and situation awareness through interactive visualization". In: *The 11th Int'l Conf. on Information Fusion*. IEEE. 2008, pp. 1–8.

[65] B. J. Rhodes, N. A. Bomberger, M. Seibert, and A. M. Waxman. "Maritime situation monitoring and awareness using learning mechanisms". In: *Conf. on Military Communications*. IEEE. 2005, pp. 646–652.

[66] R. Laxhammar. "Anomaly detection for sea surveillance". In: *The 11th Int'l Conf. on Information Fusion*. IEEE. 2008, pp. 1–8.

[67] M. Andersson and R. Johansson. "Multiple sensor fusion for effective abnormal behaviour detection in counter-piracy operations". In: *Int'l Conf. on Waterside Security*. IEEE. 2010, pp. 1–7.

[68] B. Ristic, B. La Scala, M. Morelande, and N. Gordon. "Statistical analysis of motion patterns in AIS data: Anomaly detection and motion prediction". In: *The 11th Int'l Conf. on Information Fusion*. IEEE. 2008, pp. 1–7.

[69] C. Carthel, S. Coraluppi, and P. Grignan. "Multisensor tracking and fusion for maritime surveillance". In: *The Tenth Int'l Conf. on Information Fusion*. IEEE. 2007, pp. 1–6.

[70] M. Guerriero, P. Willett, S. Coraluppi, and C. Carthel. "Radar/AIS data fusion and SAR tasking for maritime surveillance". In: *The 11th Int'l Conf. on Information Fusion*. IEEE. 2008, pp. 1–5.

[71] B. J. Rhodes, N. A. Bomberger, M. Seibert, and A. M. Waxman. "SeeCoast: Automated port scene understanding facilitated by normalcy learning". In: *Conf. on Military Communications*. IEEE. 2006, pp. 1–7.

[72] M. Vespe, M. Sciotti, F. Burro, G. Battistello, and S. Sorge. "Maritime multi-sensor data association based on geographic and navigational knowledge". In: *Radar Conf.* IEEE. 2008, pp. 1–6.

[73] S. F. Andler, M. Fredin, P. M. Gustavsson, J. van Laere, M. Nilsson, and P. Svenson. "SMARTracIn: A concept for spoof resistant tracking of vessels and detection of adverse intentions". In: *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense VIII*. Vol. 7305. Int'l Society for Optics and Photonics. 2009, 73050G.

[74] Z. Ding, G. Kannappan, K. Benameur, T. Kirubarajan, and M. Farooq. "Wide area integrated maritime surveillance: An updated architecture with data fusion". In: *Proc. of the Sixth Int'l Conf. on Information Fusion*. Vol. 2. 2003, pp. 1324–1333.

[75] E. Lefebvre and C. Helleur. "Automated association of track information from sensor sources with non-sensor information in the context of maritime surveillance". In: *Proc. of the Seventh Int'l Conf on Information Fusion*. Citeseer. 2004.

[76] J.-P. Mano, J.-P. Georgé, and M.-P. Gleizes. "Adaptive multi-agent system for multi-sensor maritime surveillance". In: *Advances in Practical Applications of Agents and Multiagent Systems*. Springer, 2010, pp. 285–290.

[77] M. Riveiro and G. Falkman. "Interactive visualization of normal behavioral models and expert rules for maritime anomaly detection". In: *The Sixth Int'l Conf. on Computer Graphics, Imaging and Visualization*. IEEE. 2009, pp. 459–466.

[78] N. Eriksson. *Predicting demand in districtheating systems: A neural network approach*. 2012.

[79] E. Dotzauer. "Simple model for prediction of loads in district-heating systems". In: *Applied Energy* 73.3-4 (2002), pp. 277–284.

[80] N. Fumo. "A review on the basics of building energy estimation". In: *Renewable and Sustainable Energy Reviews* 31 (2014), pp. 53–60.

[81] H. Wiklund. “Short term forecasting on the heat load in a DH-system”. In: *Fernwärme Int’l* 20.5-6 (1991), pp. 286–294.

[82] L. Wu, G. Kaiser, D. Solomon, R. Winter, A. Boulanger, and R. Anderson. “Improving efficiency and reliability of building systems using machine learning and automated online evaluation”. In: *The 11th Conf. on Systems, Applications and Technology*. IEEE. 2012, pp. 1–6.

[83] S. Idowu, S. Saguna, C. Åhlund, and O. Schelén. “Forecasting heat load for smart district heating systems: A machine learning approach”. In: *Int’l Conf. on Smart Grid Communications*. IEEE. 2014, pp. 554–559.

[84] T. Catalina, V. Iordache, and B. Caracaleanu. “Multiple regression model for fast prediction of the heating energy demand”. In: *Energy and Buildings* 57 (2013), pp. 302–312.

[85] K. Kato, M. Sakawa, K. Ishimaru, S. Ushiro, and T. Shibano. “Heat load prediction through recurrent neural network in district heating and cooling systems”. In: *Int’l Conf. on Systems, Man and Cybernetics*. IEEE. 2008, pp. 1401–1406.

[86] M. Sakawa, K. Kato, and S. Ushiro. “Cooling load prediction in a district heating and cooling system through simplified robust filter and multilayered neural network”. In: *Applied Artificial Intelligence* 15.7 (2001), pp. 633–643.

[87] S. Provatas. *An online machine learning algorithm for heat load forecasting in district heating systems*. 2014.

[88] S. Rongali, A. R. Choudhury, V. Chandan, and V. Arya. “A context vector regression based approach for demand forecasting in district heating networks”. In: *Int’l Conf. on Innovative Smart Grid Technologies Asia*. IEEE. 2015, pp. 1–6.

[89] K. Mařík, Z. Schindler, and P. Stluka. “Decision support tools for advanced energy management”. In: *Energy* 33.6 (2008), pp. 858–873.

[90] D. Chinese and A. Meneghetti. “Optimisation models for decision support in the development of biomass-based industrial district-heating networks in Italy”. In: *Applied Energy* 82.3 (2005), pp. 228–254.

[91] P. Bardouille and J. Koubsky. “Incorporating sustainable development considerations into energy sector decision-making: Malmö Flintränen district heating facility case study”. In: *Energy Policy* 28.10 (2000), pp. 689–711.

[92] S. N. Petrovic and K. B. Karlsson. “Danish heat atlas as a support tool for energy system models”. In: *Energy Conversion and Management* 87 (2014), pp. 1063–1076.

[93] A. Meneghetti and G. Nardin. “Enabling industrial symbiosis by a facilities management optimization approach”. In: *J. of Cleaner Production* 35 (2012), pp. 263–273.

[94] E. Brembilla and A. Sciomachen. *Design and verification of a large size district heating network by a DSS*. 1990.

[95] C. Bordin, A. Gordini, and D. Vigo. “An optimization approach for district heating strategic network design”. In: *European J. of Operational Research* 252.1 (2016), pp. 296–307.

[96] A. Sciomachen and R. Sozzi. “The algorithmic structure of a decision support system for a design of a district heating network”. In: *Computers & Operations Research* 17.2 (1990), pp. 221–230.

[97] S. S. Krishnan and R. K. Sitaraman. “Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs”. In: *IEEE/ACM Transactions on Networking* 21.6 (2013), pp. 2001–2014.

[98] D. Barbará, Y. Li, J. Couto, J.-L. Lin, and S. Jajodia. “Bootstrapping a data mining intrusion detection system”. In: *Proc. of the 2003 ACM Symp. on Applied computing*. 2003, pp. 421–425.

[99] B. Rossi, S. Chren, B. Buhnova, and T. Pitner. “Anomaly detection in Smart Grid data: An experience report”. In: *Int’l Conf. on Systems, Man, and Cybernetics*. IEEE. 2016, pp. 002313–002318.

[100] E. Hoque, R. F. Dickerson, S. M. Preum, M. Hanson, A. Barth, and J. A. Stankovic. “Holmes: A comprehensive anomaly detection system for daily in-home activities”. In: *Int’l Conf. on Distributed Computing in Sensor Systems*. IEEE. 2015, pp. 40–51.

[101] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. “A density-based algorithm for discovering clusters in large spatial databases with noise”. In: *KDD*. Vol. 96. 34. 1996, pp. 226–231.

[102] S. Guha, R. Rastogi, and K. Shim. “CURE: An efficient clustering algorithm for large databases”. In: *ACM Sigmod Record*. Vol. 27. 2. ACM. 1998, pp. 73–84.

[103] S. Guha, R. Rastogi, and K. Shim. “ROCK: A robust clustering algorithm for categorical attributes”. In: *Proc. of the 15th Int’l Conf. on Data Engineering*. IEEE. 1999, pp. 512–521.

[104] L. Ertöz, M. Steinbach, and V. Kumar. “Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data”. In: *Proc. of the 2003 SIAM Int’l Conf. on Data Mining*. SIAM. 2003, pp. 47–58.

[105] F. A. González and D. Dasgupta. “Anomaly detection using real-valued negative selection”. In: *Genetic Programming and Evolvable Machines* 4.4 (2003), pp. 383–403.

[106] X. Pan, J. Tan, S. Kavulya, R. Gandhi, and P. Narasimhan. “Ganesha: Blackbox diagnosis of mapreduce systems”. In: *ACM SIGMETRICS Performance Evaluation Review* 37.3 (2010), pp. 8–13.

[107] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. “A geometric framework for unsupervised anomaly detection”. In: *Applications of data mining in computer security*. Springer, 2002, pp. 77–101.

[108] M.-F. Jiang, S.-S. Tseng, and C.-M. Su. “Two-phase clustering process for outliers detection”. In: *Pattern Recognition Letters* 22.6 (2001), pp. 691–700.

[109] X. Wang, X. L. Wang, and D. M. Wilkes. “A minimum spanning tree-inspired clustering-based outlier detection technique”. In: *Ind. Conf. on Data Mining*. Springer. 2012, pp. 209–223.

[110] A. C. Müller, S. Nowozin, and C. H. Lampert. “Information theoretic clustering using minimum spanning trees”. In: *Joint DAGM (German Association for Pattern Recognition) and OAGM Symp.* Springer. 2012, pp. 205–215.

[111] E. Cipolla, U. Maniscalco, R. Rizzo, D. Stabile, and F. Vella. “Analysis and visualization of meteorological emergencies”. In: *Ambient Intelligence and Humanized Computing* 8.1 (2017), pp. 57–68.

[112] D. C. Montgomery. *Design and analysis of experiments*. John wiley & sons, 2000.

[113] Elektro Ljubljana. *Smart meters recorded events dataset*. 2018. URL: https://data.edincubator.eu/organization/elektro-ljubljana-podjetje-zadistribucijo-elektricne-energije-d-d.

[114] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[115] P. J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *J. of Computational and Applied Mathematics* 20 (1987), pp. 53–65.

[116] T. D. Cook, D. T. Campbell, and W. Shadish. *Experimental and quasi-experimental designs for generalized causal inference*. Houghton Mifflin Boston, 2002.

[117] B. Micenková, X.-H. Dang, I. Assent, and R. T. Ng. "Explaining outliers by subspace separability". In: *The 13th Int'l Conf. on Data Mining*. IEEE. 2013, pp. 518–527.

8

Open data for anomaly detection in maritime surveillance

Samira Kazemi, Shahrooz Abghari, Niklas Lavesson, Henric Johnson, Peter Ryman
In: International Journal of Expert Systems with Applications (40)14 (2013): pp. 5719-5729.

Abstract

Maritime surveillance has received increased attention from a civilian perspective in recent years. Anomaly detection is one of many techniques available for improving the safety and security in this domain. Maritime authorities use confidential data sources for monitoring the maritime activities; however, a paradigm shift on the Internet has created new open sources of data. We investigate the potential of using open data as a complementary resource for anomaly detection in maritime surveillance. We present and evaluate a decision support system based on open data and expert rules for this purpose. We conduct a case study in which experts from the Swedish coastguard participate to conduct a real-world validation of the system. We conclude that the exploitation of open data as a complementary resource is feasible since our results indicate improvements in the efficiency and effectiveness of the existing surveillance systems by increasing the accuracy and covering unseen aspects of maritime activities.

8.1 Introduction

Maritime surveillance is the effective understanding of all maritime activities that could impact the security, safety, economy or environment[1]. Maritime

[1] Integrating Maritime Surveillance, common information sharing environment (cise), www.ec.europa.eu/maritimeaffairs/policy/integrated_maritime_surveillance/documents/integrating_maritime_surveillance_en.pdf

transport handles over 80% of the volume of global trade[2]. Along with the development of the maritime transport system, the threats to maritime security such as illegal fishing and pollution, terrorism, smuggling activities and illegal immigration are increasing correspondingly. According to the Department of Homeland Security[3], anomaly detection is one of several techniques available for improving the safety and security in the maritime domain. Furthermore, an efficient maritime surveillance system requires a complete recognized maritime picture, which can be defined as a composite picture of maritime activities over an area of interest [1]. For national maritime sovereignty, this picture should include all activities within the 200 nautical miles wide exclusive economic zone. However, for some purposes such as the detection of illegal vessel transits, the recognized maritime picture could extend beyond this region [2]. Using today's technology, continuous tracking of all maritime activities by a single sensor is insufficient since it cannot monitor everything that happens in the surveillance area. On the other hand, there are large amounts of data in the maritime domain that are gathered from a variety of sensors, databases and information systems. Therefore, by taking advantage of all the available data sources it would be possible to obtain a complete recognized maritime picture. The maritime surveillance systems generally use closed data sources that belong to the surveillance area of each country and are obtained from a variety of sensors and databases that are only accessible by the national authorities (see Section 8.2 For detecting some of the anomalous activities such as smuggling, the maritime data beyond the surveillance area of each country are required. In order to assure security, maritime organizations in different countries need to exchange their privileged data and for this purpose they should deal with the diverse regulations of the data protection in each land. Exchanging data among countries is difficult, time-consuming and in some cases impossible because of the legislative issues. Moreover, there are activities that are neither reported to the maritime organizations, nor recorded in their data sources but these activities can be useful for surveillance purposes. The publicly accessible and reusable data that are free from the legislative issues are referred to as open data. Some of the open data sources may help in revealing previously unknown aspects of

[2] United Nations Conference on Trade and Development (UNCTAD), Review of Maritime Transport 2011, `www.unctad.org/en/Docs/rmt2011_en.pdf`

[3] National plan to achieve maritime domain awareness for the national strategy for maritime security, `www.dhs.gov/xlibrary/assets/HSPD_MDAPlan.pdf`

maritime activities. For example, there are different organizations such as ports that publish their vessel traffic data or their facility information online. In addition to the organizations, there are different online communities such as blogs, forums and social networks which provide the possibility of sharing information about maritime events. By exploiting the open data along with other confidential sources of data in the detection process, the anomaly detection can be done more wisely and the results can have more facts of interests for the maritime experts.

8.1.1 Contribution

This article contributes with a deeper understanding of open data as a complementary resource for establishing maritime surveillance operations. It provides a framework for anomaly detection based on the integration of open and closed data sources in the maritime surveillance domain. According to the framework, an anomaly detection system is developed which employs suitable algorithms to implement expert rules for detecting anomalies. Finally, this article contributes with a real-world validation of the developed anomaly detection system. The validation was performed by officers from the Swedish coastguard.

8.1.2 Outline

The remainder of this work is organized as follows: Section 8.2 reviews the background and related work regarding the open data and anomaly detection in the maritime surveillance domain. Section 8.3 and Section 8.4 present the identified open data sources and describe the case study. The framework design and implementation described in Section 8.5 and Section 8.6. Section 8.7 presents the system verification results and the validation results are shown in Section 8.8. Section 8.9 features a detailed discussion about the obtained results. Finally, Section 8.10 concludes the research with a discussion on the possible directions for future work.

8.2 Background

The idea behind open data has been established for a long time. Open data can be used in a variety of domains and can be obtained from any resource. The two major sources of open data are the open data in science and the

open data in government. The longstanding concept of open data in science tries to overcome the difficulties in the current system of scientific publishing such as the inability to access data or usage limitation that is applied by the publishers or data providers [3]. Different groups, individuals and organizations are gathered to participate in a movement toward reforming the process of scientific publication [3]. One of the outcomes of the open data movement in science is the online availability of large number of scientific datasets for the public by different organizations. As well as the open data movement in science, governments for over a decade attempt to publish government data online and make them publicly accessible, readily available, understandable and usable [4]. The sharing of government data with the public can provide openness and transparency to citizens. It can also improve the degree of participation in the society activities and the efficiency and effectiveness of the government services and the operations within and between the governments [5].

According to one estimation [6], of all information is open source, 9% is grey information (such as preprints of scientific articles, rumours in business circles, project proposals submitted to a research-funding agency, discussions with well-informed specialists, etc.), 0.9% is secret and 0.1% is non-existent information (i.e. the information you have, but you are not aware of it). Considering the large ratio of the open data sources, there should be a great value in using them in different domains. In the maritime surveillance systems, the majority of the exploited data are obtained from the confidential sources. However, in recent years the new concept of the Web, which takes the network as a platform for information sharing, interoperability and collaboration, has created new sources of data for maritime surveillance. There are organizations and communities that provide their maritime related data online and make them accessible for the public. Therefore, it would be beneficial for the maritime surveillance systems if they can take advantage of the open data to increase the safety and security in their surveillance area.

8.2.1 Terminology

Anomaly detection is widely used in the areas such as video surveillance, network security and military surveillance. Chandola, Banerjee and Kumar [7] define AD as: *The problem of finding patterns in data that do not conform to expected behavior.*

Depending on the domain of study, the non-conforming patterns are called by different names such as anomalies, outliers, exceptions, etc. In the maritime surveillance domain, these non-conforming patterns are referred as anomalies. Defense research and development Canada [8] provides the following definition for the term anomaly in the context of the maritime surveillance domain: *Something peculiar (odd, curious, weird, bizarre, atypical) because it is inconsistent with or deviating from what is usual, normal, or expected, or because it is not conforming to rules, laws or customs.*

The term *open data* refers to the idea of making data freely available to use, reuse or redistribute without any restriction. The open data movement follows the other open movements such as *open access* and *open source*. According to the Open Knowledge Foundation[4], a community based organization that promotes open knowledge (whether it is content, data or information-based), an open work should be available as a whole, with a reasonable reproduction cost, preferably downloading via the Internet without charge and in a convenient and modifiable form. Furthermore, it should be possible to modify and distribute the work without any discrimination against persons, groups, fields or endeavour. In the scope of this study, the open data term refers to the publicly available data that may or may not require free registration.

8.2.2 Related Work

In recent years, the number of studies that address the use of anomaly detection in the maritime surveillance domain is increasingly growing. Anomaly detection techniques are divided into two groups, namely data-driven and knowledge-driven approaches. There are a couple of works that proposed knowledge-based anomaly detection systems with different representation techniques and reasoning paradigms such as rule-based, description logic and case-based reasoning [9–11]. A prototype for a rule-based expert system based on the maritime domain ontologies was developed by Edlund, Gronkvist, Lingvall, and Sviestins [12] that could detect some of the anomalies regarding the spatial and kinematic relation between objects such as simple scenarios for hijacking, piloting and smuggling. Another rule-based prototype was developed by Defence R&D Canada [8, 13]. The aforementioned prototype employed various maritime situational facts about both the

[4] Open definition, `opendefinition.org/okd/`

kinematic and static data in the domain to make a rule-based automated reasoning engine for finding anomalies. One of the popular data-driven anomaly detection approaches is the Bayesian network [14–16]. Johansson and Falkman [15] used the kinematic data for creating the network; however, in the work that was done by Fooladvandi et al. [14] expert's knowledge as well as the kinematic data was used in the detection process. Moreover Lane et al. [16] presented the detection approaches for five unusual vessel behaviors and the estimation of the overall threat was performed by using a Bayesian network. Unsupervised learning techniques have been widely used for data-driven anomaly detection such as Trajectory Clustering [17], self organizing map [18] and fuzzy ARTMAP neural network [19]. Some statistical approaches, such as Gaussian mixture model [20], hidden Markov model [21], adaptive kernel density estimator [22] and precise/imprecise state-based anomaly detection [17] have been used in this context. The majority of the works that have been done in the context of anomaly detection only used transponder data from the Automatic Identification System (AIS).

There are a number of studies that employed data fusion techniques to fuse data from different sensors in anomaly detection systems [23–26]. In these studies, the surveillance area was restricted to the coastal regions and the combination of data from AIS, synthetic aperture radar, infra-red sensors, video and other types of radar was used in the fusion process to obtain the vessel tracks. Furthermore, there are some other works that focused on the fusion of both sensor and non-sensor data [14, 27–31]. For example, Lefebvre and Helleur [29] and Riveiro and Falkman [31] treated the expert's knowledge as the non-sensor data. Riveiro and Falkman [31] introduced a normal model of vessel behavior based on AIS data by using self organizing map and a Gaussian mixture model. According to the model, the expert's knowledge about the common characteristic of the maritime traffic was captured as if – then rules and the anomaly detection procedure was supposed to find the deviation from the expected value in the data. Lefebvre and Helleur [29] used radar data with user's knowledge about the vessels of interests. The sensor data were modelled as track and the non-sensor data were modelled as templates. The track-template association was done by defining mathematical models for tracks and using fuzzy membership functions for association possibilities. Mano [30] proposed a prototype for the maritime surveillance system that could collect data from different types of sensors and databases and regroup them for each

vessel. Sensors like AIS, high frequency surface wave radar and classical radars and databases such as environmental database, Lloyd's Insurance and TF2000 Vessel database were included in this prototype. By using multi-agent technology an agent was assigned to each vessel and anomalies could be detected by employing a rule-based inference engine. When the combination of anomalies exceeded a threshold, vessel status was informed to the user as an anomaly. The work presented by Ding et al. [28], proposed the architecture of a centralized integrated maritime surveillance system for the Canadian coasts. Sensors and databases included in this architecture were: high frequency surface wave radar, automatic dependant surveillance reports, visual reports, information sources, microwave radar, and radar sat. A common data structure was defined for storing data that were collected from different sensors. Andler et al. [27], also described a conceptual maritime surveillance system that integrated all available information such as databases and sensor systems (AIS, long-range identification and tracking, intelligence reports, registers/databases of vessels, harbours, and crews) to help user to detect and visualize anomalies in the vessel traffic data in a worldwide scale. Furthermore, the authors suggested using open data in addition to other resources in the fusion process.

In conclusion, the main focus of the studies that have been done in the context of anomaly detection in the maritime surveillance domain was related to using sensors data and mainly the AIS data to find anomalies in the coastal regions. Detection of some suspicious activities such as smuggling requires vessel traffic data beyond the coastal region. Maritime authorities in each country have overall information of maritime activities in their surveillance area. But exchanging information among different countries is a complicated procedure because of the diverse regulation of data protection in each land. Therefore, using data sources that are free from legislative procedures can be a good solution for providing information that belongs to the regions outside the land territory. Furthermore, all the information about maritime activities is not recorded in the authorities' databases or reported to them. On the other hand, there are numerous open data sources consists of different websites, blogs and social networks that can be useful for observing the hidden aspects of maritime activities. Hence, this article will investigate the potential open data sources for maritime activities and exploit them to build an anomaly detection system. The aim of this system is to provide complementary decision support for coastguard operators when

they analyze traditional closed data sources.

8.3 Open Data in Maritime Surveillance

To obtain the applicable open data for anomaly detection, the first step is initiated by reviewing the information resources document[5] provided by the International Maritime Organization. This organization is the United Nations' specialized agency with responsibility for the safety and security of shipping and the prevention of marine pollution by vessels. The document introduces 29 governmental and intergovernmental organizations that work in different fields related to the maritime surveillance domain such as maritime safety, prevention of pollution from vessels, liability and insurance issues, shipping information, etc. All these 29 organizations' websites and the links provided by each of them are investigated and a list of online data sources is prepared. The obtained open data sources provide AIS data, information about vessel characteristics, ports, maritime companies, suppliers, weather, etc. Moreover, in the process of finding open data sources, an attempt is made to obtain sources of data that are related to the Baltic region and mostly Sweden by use of the previously observed data sources and also common search engines. This extensive collection of open data sources for the maritime surveillance domain is available for download[6].

8.4 Case Study

The case study is the employed research method in this article. The two important sources of information about maritime anomalies are reports of the workshops that were held in Canada [8] and Sweden [27, 32]. In these two workshops attendees were experts in the maritime domain and a variety of maritime anomalies were identified.

According to the identified anomalies by the two workshops, a list of some potential maritime anomalies that can be detected by use of the available open data sources was prepared in the preliminary work of this

[5] Information resources on maritime security and ISPS code, `www.imo.org/knowledgecentre/informationresourcesoncurrenttopics/maritimesecurityandispscode/documents/information-resources-on-maritime-security-and-isps-code.pdf`

[6] Open maritime-specific data collection, `http://www.bth.se/com/nla.nsf/sidor/resources`.

study. Then, in a meeting with representatives of the Swedish coastguard the types of anomalies that are of high interest for the coastguard operators and the possibility of using open data for anomaly detection were discussed. During the meeting, the prepared list of anomalies was reviewed. Coastguard operators were asked about the possibility of the historical occurrence, and their degree of interest, for each anomaly. As an outcome of the meeting a number of scenarios were created and based on these scenarios, 11 expert rules were defined.

The first scenario refers to the anomalies related to the vessel static information such as name, owner, International Maritime Organization number, dimensions, type and the status (in service or laid up). For example, sailing a vessel with a draught of 22 meters over an area with a 9 meter depth or observing a vessel that should be laid up or changing the name or the owner of a vessel during its voyage indicate the existence of suspicious activities.

The second scenario is related to the prior arrival notification for vessels. Vessels should inform their arrival time to the ports at least 24 h in advance. Each port also provides an online timetable for the incoming vessels. Therefore, any mismatch between the reported AIS data regarding the destination or the arrival time of a vessel and the destination port timetable needs to be checked by the coastguards.

The third scenario is related to ordering pilots. Usually, large vessels because of their size and weight need to be guided by pilots through dangerous and congested waters. Therefore, vessels need to submit their request for a pilot and also inform the destination port. However, in some cases vessels order a pilot without informing the port. Such situations should be investigated.

The case study in this article comprises scenarios two and three, which are conducted in close collaboration with operators from the Swedish coastguard. The aim of the case study is to investigate the potential of open data about maritime activities (vessels, ports, and so on) as a complement to the closed data sources that are already used by the Swedish coastguard. They key questions posed in this study are:

- How can open data complement closed data for anomaly detection in the maritime surveillance domain?

- What is the positive and negative impacts of the open data sources? (that is, what is the increase in true negatives and positives in comparison to the increase in false negatives and positives?)

In the next meeting, the scenarios and the rules are presented to the representatives of the Swedish coastguard and they are asked to comment or suggest new scenarios or rules. By getting the final approval from the coastguard experts, one new rule (rule number 5) is added to the list. Table 8.1 shows the admitted rules by the experts. These identified maritime anomalies can be detected by use of AIS data, vessel traffic timetables in ports and pilots websites and the vessel characteristic data that are available in data sources such as Lloyd's. A capitalized name is provided to each anomaly that can be detected by the rules, and for the remainder of this article the anomalies will be referred to by these names.

Table 8.1: The identified anomalies that can be detected by open data (confirmed by the Swedish Coastguard)

No.	Expert rules	Anomaly
1	If a vessel destination does not exist in the port schedule then anomaly.	VESSEL_NOT_INFORMED_PORT (A1)
2	If a vessel ETA does not match with the port ETA for the vessel then anomaly.	ARRIVAL_TIME_MISMATCHED (A2)
3	If a vessel entered a port without informing the port then anomaly.	VESSEL_ENTERED_PORT_WITHOUT_NOTICE (A3)
4	If a vessel has requested a pilot but has not used the service then anomaly.	VESSEL_NOT_USED_PILOT (A4)
5	If vessel A which normally travels between ports X and Y, suddenly goes to port Z then anomaly.	UNUSUAL_TRIP_PATTERN (A5)
6	If a vessel has not left a port according to the port schedule then anomaly.	VESSEL_NOT_LEFT_PORT (A6)
7	If a vessel exists in a port schedule but it has not entered the port then anomaly.	VESSEL_NOT_ENTERED_PORT (A7)
8	If a vessel does not exist in the port schedule and the vessel has requested a pilot then anomaly.	VESSEL_ORDERED_PILOT_AND_NOT_INFORMED_PORT (A8)
9	If a vessel has moored in a port and has been observed somewhere else then anomaly.	VESSEL_MOORED_IN_PORT (A9)
10	If vessel A has not entered a port according to the port schedule instead vessel B enters the port at the same time slot then anomaly.	WRONG_VESSEL_ENTERED (A10)
11	If a vessel with the laid up status has been observed somewhere else then anomaly.	VESSEL_LAID_UP (A11)

Note. ETA = estimated time of arrival.

8.5 Framework Design

A new maritime surveillance framework and expert-based decision support system is presented in this article. The Open Data Anomaly Detection System (ODADS) is designed for traffic monitoring and detecting anomalies in the maritime domain by using open and closed data sources. Fig. 8.1 depicts the ODADS architecture. The framework is designed to be generalizable to similar applications in other domains; that is, for applications where the objective is to identify anomalous behavior through semi-automatic methods. The proposed framework is designed to provide decision-support based on knowledge-engineering-based or knowledge-discovery-based methods. It is focused on the extraction of information from open data sources. The setup of any implementation of the framework depends largely on the problem at hand. ODADS consists of three core modules:

- Data Collector
- Anomaly Detector and,
- Display Client

The Data Collector module is responsible for collecting open data from the Internet, and for preprocessing and storing the data in the system database. The data can be related to vessel traffic (such as AIS reports, ports and pilots timetables), vessel characteristics, ports equipments and facilities, companies that are involved in maritime activities, news or reports about maritime events and activities available in different social media platforms (such as blogs and social networks), and so on. The Data Store comprises a set of databases that contain data belonging to different types of sensors, authorized databases and open data sources. The data in the Data Store can be fused or integrated before being used in the detection process. When the Data Collector completes its task, the Anomaly Detector becomes available. The Anomaly Detector module analyses the available open and closed data and detects possible anomalies by using both knowledge-driven and data-driven techniques. Different anomaly detection techniques are employed due to the distinct nature of anomalies and the complexity of the environment in the maritime surveillance domain. Previously known anomalies can be detected by knowledge-based techniques but in real-world situations it is desirable that an anomaly detection system can detect previously

unseen anomalies as well. One of the potential benefits of using data-

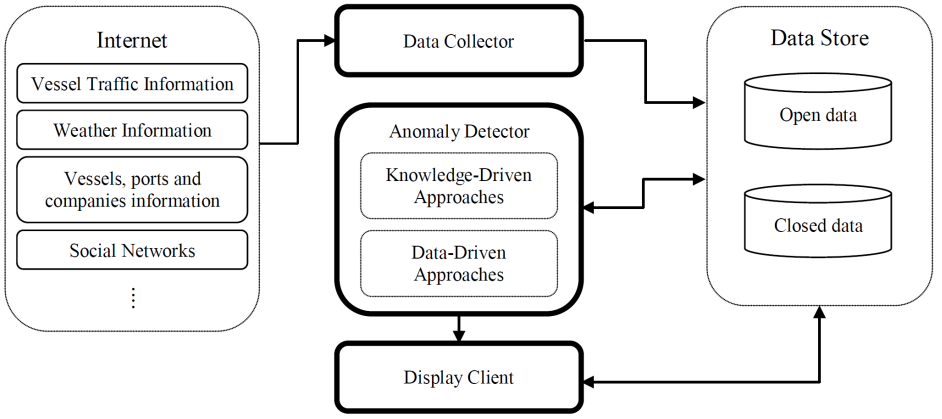


Figure 8.1: The Open Data Anomaly Detection System (ODADS) architecture. The Data Collector module collects data from the Internet and stores them in the database. The Anomaly Detector module detects anomalies by taking advantage of different techniques. The Display Client module displays the detected anomalies to the user and enables system-user interaction.

driven methods such as machine learning algorithms is the possibility of detecting such unseen anomalies. However, it is difficult to evaluate how well today's data-driven systems manage to detect previously unseen cases. The proposed system in this article is deterministic and completely expert-based but our proposed framework general enough to allow data-driven detection methods in other applications. The Display Client module is the user interface of the system. This module represents the cognitive refinement level (Level 5) of the Joint Directors of Laboratories model. It is argued that the effectiveness of a system can be affected by the way that the system produced information is comprehended by the human user [33]. The cognitive refinement process involves traditional human computer interaction utilities such as geographical display or advanced methods that support functionalities such as cognitive aids, negative reasoning enhancement, focus/defocus of attention and representing uncertainty. While designing the user interface, the six principles of user interface design that are based on the usage-centered design approach are considered. These six principles are: structure, simplicity, visibility, feedback, tolerance, and reuse [34].

8.6 Implementation

ODADS is implemented by taking advantage of the identified maritime anomalies and the obtained open data sources. To limit the scope, four types of vessels (passenger, ferry, cargo, and tanker) are considered. Other types of vessels (such as fishing and sailing vessels) are omitted. Secondly, the rule related to the vessel static information is ignored. Furthermore, the WRONG_VESSE_ENTERED anomaly is excluded due to its complexity. Moreover, in further collaboration with the coastguard representatives during the implementation phase, a new type of anomaly is proposed. This anomaly is called UNDER_SURVEILLANCE_VESSEL and occurs when a vessel of interest has any of the A1–A9 anomalies and the vessel exists in the vessels blacklist.

8.6.1 Data Description

The required vessel traffic data can be obtained from AIS reports and ports and pilots timetables. The surveillance area is restricted to the north of the Baltic Sea and a part of the Gulf of Finland, the regional area between three European countries Sweden, Finland and Estonia. Fig. 8.2 shows the surveillance area where the geographic coordinates lie between latitudes 58.49° – 60.24° N and longitude 16.19° – 25.00° E. This region is one of the high-traffic regions in the Baltic Sea and is surrounded by the four highly used ports. The selected ports are: Stockholm group (Stockholm, Kapellskär and Nynäshamn) and Norrköping ports in Sweden, Helsinki port in Finland and Tallinn port in Estonia. Due to inaccessibility to the raw AIS data (a closed source) in the surveillance area, the AIS reports that are provided by the *MarineTraffic.com* website are exploited. These reports consist of both static and dynamic types of data for each vessel during its voyage such as name, type, year built, flag, call sign, maritime mobile service identity, International Maritime Organization identification number, origin, destination, Estimated Time of Arrival, speed (maximum and average), position (longitude and latitude), and heading. The pilot data belong to the Stockholm pilotage area in Sweden. Moreover, a common data representation format for ports and pilots data is defined that contains vessel name, vessel type, origin, destination, company name, vessel status and arrival/departure time. Table A2 in the appendix provides more details about these sources.

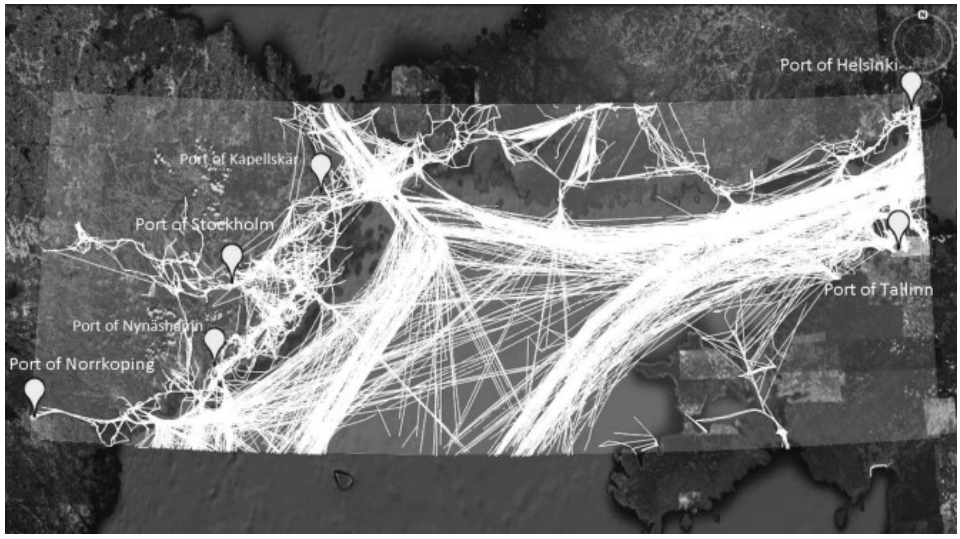


Figure 8.2: The surveillance area and the vessels tracks extracted from AIS datafor 6 days. The area is restricted to the north of the Baltic Sea and part of the Gulf of Finland. Ports from left to right are Norrköping, Nynäshamn, Stockholm, Kapellskär, Helsinki and Tallinn (The image is generated by Google Earth).

8.6.2 Detection Methods

After investigating the nature of the anomalies and the potential techniques, it is determined that except for the UNUSUAL_TRIP_PATTERN anomaly, detection of other anomalies can be done by performing a search in the data for finding the desired match. If the match is not found then the vessel would be marked as an anomaly. Using exact string matching techniques for comparing vessel information from different data sources is inapplicable due to the potential errors that might occur because of different notations or human operator mistakes during data entry. Therefore, a metric should be used for measuring the degree of similarity between two vessels from different sources. After investigating the performance of different string matching techniques on the available data, the *JaroWinkler*[7] metric is chosen. For two strings if the JaroWinkler distance is less than or equal to a predefined threshold then the two strings are considered similar.

[7] JaroWinkler (the variant of Jaro) measures the number and order of common characters in the two strings and also the number of transposition that needs to change one of the strings to the other.

Detection of the UNUSUAL_TRIP_PATTERN anomaly requires data-driven approaches such as machine learning or statistical techniques. Unlike the other anomalies, detection of this anomaly requires a history of vessel traffic data for training the system. For this reason, data related to six months (September 15, 2011-March 15, 2012) of vessel traffic in the surveillance area are gathered. By monitoring the activities during this period, the system will be able to find the normal pattern of vessels trips in the area of interest. For detecting this anomaly a simple statistical approach is used. A look up table is created and for each vessel the number of times that the vessel travels between two different ports is stored. For each vessel, if the frequency of travelling between its origin and destination is less than a predefined threshold (which is 2 in the current implementation) then the vessel will be reported as an anomaly.

There are situations that multiple anomalies can occur in the same time for a specific vessel. The combinations of anomalies that don't have any features in common are defined as new types of anomalies. For instance, a vessel has not informed its arrival to the destination port and its trip to the port is not common, in such situation a new type of anomaly UNUSUAL_TRIP_PATTERN_AND_VESSEL_NOT_INFORMED_PORT is defined.

8.7 System Verification

To ensure that ODADS works properly, the system is tested manually with both real and manipulated data. The tests are performed during the implementation and also after completing the system. At first, a number of vessels with different types of anomalies are inserted to the real collected data to check whether all types of anomalies can be detected by ODADS. Then, the system is run for a period of time and the detected anomalies are checked manually against the available data to make sure about their correctness. A screenshot of the ODADS visualization of the maritime environment is shown in Fig. 8.3. During the test phase the Anomaly Detector module is updated and some of the detection conditions are narrowed down. The process is repeated until the system can detect all the anomalies correctly. Furthermore, before using ODADS in real-world situations, it is important to figure out to what extent the results of the system are accurate. For this reason, an experiment is conducted to measure the accuracy. Accuracy is

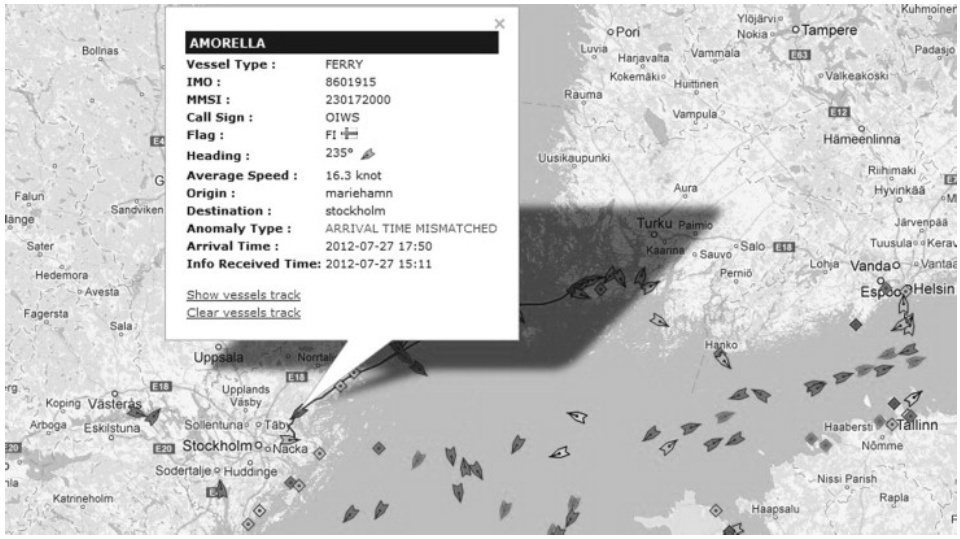


Figure 8.3: A screenshot of the ODADS visualization of the maritime environment.

the degree to which the estimates or measurements of a quantity correctly describe the exact value of that quantity. In other words, accuracy is the proportion of true results in the population. To evaluate the system accuracy, the number of True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN) are needed. Accuracy is calculated by the following formula [35]:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8.1}$$

The first step in designing the experiment is to identify the population. The population consists of the vessel traffic data in the surveillance area. Since the population is too large and it is impossible to look into all members manually to count the number of TP, FP, TN and FN, a sample should be taken from the population. The sampling frame is the vessel traffic data related to AIS, ports and pilots in the surveillance area, which are provided by ODADS. Due to the high volume of traffic through the surveillance area, it is expected that the majority of anomalies can be observed in one week execution of the system. Therefore, one week of vessel traffic data from April, 2012 is used as the sample frame (Table A3 and Table A4 in the appendix present some information about the vessels traffic and detected anomalies during this week). A two stage random sampling is used in order

to have unbiased and independent samples. In the first stage, a simple random sampling without replacement is done for selecting the time slots that ODADS attempts to collect and analyse the data. After selecting the time slots, the corresponding data for each time slot will be selected by a stratified sampling. Three strata are defined according to the type of vessels: ferry and passenger, cargo and tanker vessels. Selection of vessels is also limited to the vessels that are originated from or targeted to the four particular ports. The total number of time slots in the sample frame is 835. This means that on average, ODADS collects data 139 times a day. In the first stage of sampling a random timeslot is selected for each day, which results in 7 time slots for one week. Then, by considering the described limitation in the selection process, the average number of entire data in a selected time slot is about 100 records. Among these records, 30 records are selected by stratification. Almost 73% of the vessels in each time slot are moored. Since the majority of the anomalies are related to the vessels trips, a limitation on the number of moored vessels in the samples is defined. In this way, it can be possible to check more anomalies in the evaluation process. The second stage of sampling is repeated by taking into consideration that the number of moored vessel in the sample cannot exceed from the half of the sample size (in this case, 15).

After carrying out the sampling, all the samples are checked against the primary identified anomalies ($A_1 - A_9$). To compute the number of TP, FP, TN and FN, a confusion matrix is created based on the nine classes of anomalies and the normal class (Table 8.2). According to the matrix, the accuracy of the system is: $17 + 192/17 + 192 + 0 + 1 = 0.99$. The existing FN for the ARRIVAL_TIME_MISMATCHED anomaly is due to the wrong provided AIS data by the vessel or possibly the *MarineTraffic.com* website and also the limitation of the system for considering all conditions. In this case, the vessel arrival time belongs to a couple of days before the current date and for this reason it is ignored by the system. However, it is quite possible to handle such situations if additional sources of AIS data are available.

8.8 System Validity

The validation was made by an officer at the coastguard Headquarters office in Karlskrona, Sweden for four weeks (April 23, 2012–May 18, 2012), at

Table 8.2: Confusion matrix for the nine classes of anomalies and the normal class

		Predicted class										
		A1	A2	A3	A4	A5	A6	A7	A8	A9	Normal	Total
Actual class	A1	6	-	-	-	-	-	-	-	-	-	6
	A2	-	7	-	-	-	-	-	-	-	1	8
	A3	-	-	-	-	-	-	-	-	-	-	0
	A4	-	-	-	-	-	-	-	-	-	-	0
	A5	-	-	-	-	3	-	-	-	-	-	3
	A6	-	-	-	-	-	1	-	-	-	-	1
	A7	-	-	-	-	-	-	-	-	-	-	0
	A8	-	-	-	-	-	-	-	-	-	-	0
	A9	-	-	-	-	-	-	-	-	-	-	0
	Normal	-	-	-	-	-	-	-	-	-	192	192
	Total	6	7	0	0	3	1	0	0	0	193	17

any time during working hours (08:00–17:00). The officers are supposed to evaluate the detected anomalies by checking them against the available data in the systems and data sources that are used during the normal operational activities at the coastguard. They are asked to provide weekly report about their evaluation results in order to decrease the possible malfunctioning of the system and the validation process. A detected anomaly for a vessel is true if it can be confirmed by the available data sources at the coastguard and consequently it is false if the authorized data sources provide any information that declines the detected anomaly. No further assessment is done regarding the classification of the detected anomalies to true and false alarms.

The sea monitoring system that is used by the coastguard officer is called SJöBASIS[8]. SJöBASIS aggregates the maritime data from different systems and agencies with the aim of improving the efficiency of maritime surveillance. In SJöBASIS, he required data that contain vessel position, speed, heading, arrival/departure time and trip, are obtained from the following sources SafeSeaNet[9], SjöC, local AIS and HELCOM AIS[10]. The officer checks the validity of anomalies according to the priority that each anomaly has for him. During the four-week validation period, ODADS is used at the coastguard for 12 working days and in total 76 of the detected anomalies are evaluated. Table 8.3 presents the validation results. Among the evaluated anomalies, there are a number of anomalous vessels

[8] www.kustbevakningen.se/sv/granslos-samverkan/sjoovervakningsuppdraget/samverkan-sjoinformation/

[9] www.emsa.europa.eu/operations/maritime-surveillance/safeseanet.html

[10] www.helcom.fi/BSAP/ActionPlan/en_GB/SegmentSummary/

Table 8.3: Validation results of the Coastguard

Anomaly	Alarms		
	True	False	Not Checked
VESSEL_NOT_INFORMED_PORT	7	3	4
ARRIVAL_TIME_MISMATCHED	19	5	3
VESSEL_NOT_USED_PILOT	2	-	-
UNUSUAL_TRIP_PATTERN	7	6	-
VESSEL_NOT_LEFT_PORT	1	1	-
VESSEL_ORDERED_PILOT_AND_NOT_INFORMED_PORT	2	-	-
VESSEL_MOORED_IN_PORT	-	3	-
UNDER_SURVEILLANCE_VESSEL	1	-	-
UNUSUAL_TRIP_PATTERN_AND_VESSEL_NOT_ INFORMED_PORT	5	1	-
UNUSUAL_TRIP_PATTERN_AND_VESSEL_ ARRIVAL_TIME_ MISMATCHED	4	-	-
UNUSUAL_TRIP_PATTERN_AND_VESSEL_ORDERED_PILOT_ AND_NOT_INFORMED_PORT	-	1	-
VESSEL_ORDERED_PILOT_AND_NOT_INFORMED_PORT_ AND_VESSEL_NOT_USED_PILOT	1	-	-
Total count	49	20	7
Total count (%)	64.47	26.32	9.21

that remain unchecked due to a lack of corresponding data in the coastguard systems. A large number of detected anomalies are related to the ARRIVAL_TIME_MISMATCHED anomaly that in many cases can be due to the inconsistent time formats in different data sources and various settings in the AIS transmitters. In these cases, the detected anomalies by ODADS are correct detections based on the available data but the real-world situation that the data are used represent is considered normal. For example, a vessel that is going from Helsinki to Stockholm is reporting its arrival time according to the Finland local time instead of using the coordinated universal time format. Therefore, this artificial time difference results in that the arrival time reported by the vessel does not match the expected arrival time of the vessel at the destination port, which leads to set the ARRIVAL_TIME_MISMATCHED anomaly for that vessel. In addition to the comparative analysis in the validation process, modus operandi of using an Anomaly Detector that takes advantage of open data is investigated. The coastguard officer uses an analysis tool[11] to analyse the ODADS excel reports and draw conclusions regarding the modus operandi of the system in the emergency situations that can have impact on the maritime surveillance operations. One of the possible analyses of the system reports can be the investigation of vessels with multiple anomalies. Here are some examples of

[11] IBM i2 Analyst's Notebook, `www.i2group.com/us/products/analysis-product-line/ibm-i2-analysts-notebook`

the vessels with multiple anomalies. A cargo vessel has recurring anomalies related to the arrival/departure time, trip and notification to the port. From the types of anomalies that are detected for this vessel, it can be concluded that the vessel behavior points to a higher threat concerning customs and border. For a passenger vessel the following anomalies are often detected: VESSEL_NOT_INFORMED_PORT, ARRIVAL_TIME_MISMATCHED, VESSEL_NOT_ENTERED_PORT and VESSEL_NOT_LEFT_PORT. These anomalies may happen because of inaccurate or wrong provided data by the vessel. The conclusion and modus operandi for this vessel is that the duty officer will contact the vessel to highlight the importance of submitting accurate information. In real-world situations, the occurring anomalies such as VESSEL_NOT_ENTERED_PORT or VESSEL_NOT_LEFT_PORT for a passenger vessel can be related to serious issues such as an accident and it will result in increased scrutiny for that vessel. It is also possible to look into the relation between the types of vessels and the detected anomalies. Such assessment can be used for strategic and risk analysis. For tanker vessels the most common anomaly is VESSEL_NOT_INFORMED_PORT. This anomaly has a high priority for emergency preparedness for accidents involving tankers. Tankers with UNUSUAL_TRIP_PATTERN anomaly are a potential risk to other vessels and can cause accidents. From a risk assessment point of view the combination of this anomaly with the VESSEL_NOT_ENTERED_PORT or VESSEL_NOT_USED_PILOT anomalies can lead to high-risk situations. The most occurring anomaly for ferries and passenger vessels is ARRIVAL_TIME_MISMATCHED. In several cases this anomaly is detected incorrectly because of the wrong reported arrival time; however, this anomaly has great importance for the authorities to plan their operations regarding ferries and passenger vessels arrivals effectively. The most serious anomaly for ferries and passenger vessels is VESSEL_NOT_LEFT_PORT and the authorities should suspect that some form of accident or difficulty is arising regarding to the departure of the vessels. For cargo vessels, the most recurring anomalies are VESSEL_NOT_INFORMED_PORT and ARRIVAL_TIME_MISMATCHED. However, some of the ARRIVAL_TIME_MISMATCHED anomalies are false alarms because of the incorrect data. The VESSEL_NOT_INFORMED_PORT anomaly is important for ports security and safety. The prior notification to the ports is obligatory for vessels, but the fine for breaking this rule is negligible which lets the vessels that are involved in illegal activities such as smuggling disobey this rule. The most serious anoma-

lies for the cargo vessels are the VESSEL_NOT_ENTERED_PORT and UNDER_SURVEILLANCE_VESSEL anomalies.

Furthermore, looking into the most frequent anomalies for different ports will assist the maritime authorities to make their decisions more efficiently. According to the report analysis, the most frequent anomaly in Stockholm and Nynäshamn ports is ARRIVAL_TIME_MISMATCHED, in port of Kapellskär is VESSEL_NOT_INFORMED_PORT and in Norrköping port is UNUSUAL_TRIP_PATTERN. One possible conclusion from the most popular anomalies for the ports is that the port authorities should be informed of the divergence in the traffic flow and the operational management functions in order to plan and allocate resources efficiently. On the other hand, in some cases the anomalies are too commonly occurring for a port because of the inaccurate provided data and they can be disregarded by the officer.

The received feedback from the coastguard representatives during the validation process indicates that ODADS complements the closed sources and assists the human operator in gaining a better understanding of the ongoing maritime activities. The representatives believe that the ODADS results are reliable and the quality of the open data that are used is good and can be used in real-world situations. The functionality, usability, and visualization tools in ODADS provide a simple and intuitive system for coastguard operators. In addition to illustrate the vessel traffic data in a simple, clear, and informative way, ODADS can provide statements about the anomalies and its statistical reports are beneficial when the authorities and freight companies conduct strategic analysis of maritime traffic and risk assessment. Finally, the capability of automatic detection of anomalies based on open data is considered a valuable asset to the coastguard.

8.9 Discussion

Taking advantage of anomaly detection systems will assist authorities to tighten security in the maritime surveillance domain. There are a number of studies which focused on developing anomaly detection systems by using knowledge-driven and data-driven approaches. For instance, Defence research and development Canada [8, 13] developed a rule-based prototype for anomaly detection by exploiting maritime situational facts about both kinematic and static data of the domain. Edlund et al. [12] ddeveloped another

prototype for a rule-based expert system to detect the anomalies regarding spatial and kinematic relation between objects. Riveiro and Falkman [31] proposed using a combination of data-driven and knowledge-driven approaches to detect anomalies by use of a normal model of vessel behavior based on AIS data and experts rules. In the majority of studies that addressed anomaly detection, the exploited data for the anomaly detection process were obtained from closed data sources and there is a lack of investigation on using open data sources for anomaly detection in the maritime surveillance domain. Therefore, in this article ODADS is implemented by employing expert rules to investigate the potential open data as a complement to the closed data for anomaly detection in the maritime surveillance domain.

The validity of the system is evaluated in real-world situations by the experts from the Swedish coastguard. Despite the inaccurate nature of open data and by considering the fact that only open data sources are used in the system, the high degree of true alarms (64.47%) in the validation process admits the validity of the system outcomes. Furthermore, there are no corresponding data in the authorized databases for 9.21% of the evaluated anomalies by the coastguard. This fact refers to a potential information gap in the closed data sources. However, the considerable number of false alarms (26.32%) for a surveillance system is still unsatisfactory. The number of false alarms indicates the difference between the accuracy of the system and the validity of the results. Even though the data that are used in ODADS are obtained from relatively trusted data sources such as ports, the false alarms occur mostly because of data inaccuracies. The open data that are exploited by ODADS suffer from these errors due to human operator mistakes, irregular data updates, data update latencies, and incompatible data formats. These are critical issues that unfortunately seem to concern many open data applications. In ODADS, there are situations where a detected anomaly disappears in the next periods of system execution because of the arrival of revised and corrected data. Frequent occurrences of false alarms distract the operator's attention from real anomalies in the surveillance area.

To decrease the false alarms in ODADS, the main solution is to integrate open and closed data, which can cover the lack of information or inaccuracy in the open data. In addition, considering a probability for the detected anomalies can decrease the number of false alarms. This would be possible by analysing the history of vessels behavior as well as the current situation and defining a probability threshold to omit the anomalies that have a lower

probability than the threshold. Furthermore, having extra information regarding vessels such as crew and cargo information, can affect the probability of being a real anomaly for a specific vessel. For example, if a vessel has the ARRIVAL_TIME_MISMATCHED anomaly and it has a crew member with a criminal record or a special cargo, then there is a possibility that the vessel is stopped somewhere to exchange something. Therefore, in such situation, the probability of being a true anomaly is high. According to the validation results, the UNUSUAL_TRIP_PATTERN anomaly creates the majority of false alarms. This is due in part to the statistical approach that is used for detecting this anomaly and also the wrong origin and destination information that the vessels provide. The lookup table that is created for storing the frequency of the trips between different places is not updated periodically. While populating the table, the ports timetables are used which can be incomplete. An alternative detection approach can be to use machine learning techniques, which attempt to detect anomalies according to the pattern of movements for individual vessels instead of the reported trip data by the vessels or ports.

The proposed framework is generalizable to similar applications in other domains due to its modularized and general design. This case study has investigated the potential of open data as a complement to closed data for anomaly detection in the maritime surveillance domain. The primary stakeholders in the case study are human operators from the Swedish coastguard. Since the main purpose of the case study was to focus analysis and investigation on two defined scenarios, it is not possible to draw conclusions about the generalizability of the results. Further investigations can shed light on this generalizability by conducting large-scale trials of the implemented expert-based system across additional scenarios and more maritime surveillance areas.

In local areas such as the surveillance area in this article, mainly because of large amount of quality assured data and the limited size of the surveillance area, it is easier for the maritime authorities to track and control the vessels activities. Therefore, the use of open AIS data in this region is not required and it should be prohibited to decrease the negative impacts of open data on the system results. On the other hand, when the vessel information beyond the exclusive economic zone is required, the value of open data becomes more obvious.

8.10 Conclusion and Future Work

This article investigated the potential open data as a complementary resource for anomaly detection in the maritime surveillance domain. A framework for anomaly detection was proposed based on the usage of open data sources along with other traditional sources of data. According to the proposed anomaly detection framework and the algorithms for implementing the expert rules, the Open Data Anomaly Detection System (ODADS) was developed. The validity of the results was investigated by the subject matter experts from the Swedish coastguard. The validation results showed that the majority of the ODADS evaluated anomalies were true alarms. Moreover, a potential information gap in the closed data sources was observed during the validation process. Despite the high number of true alarms, the number of false alarms was also considerable that was mainly because of the inaccurate open data. This article provided insights into the open data as a complement to the common data sources in the MS domain and is concluded that using open data will improve the efficiency of the surveillance systems by increasing the accuracy and covering unseen aspects of maritime activities.

In the future, it is important to investigate how the open data sources in the maritime domain can be used in a global perspective. In this article, the surveillance area was limited to a local area which is fully covered by the authorities' data sources. When the data beyond the exclusive economic zone are needed, it is more valuable to use open data sources. By taking advantage of the subject matter experts' knowledge about maritime surveillance, it would be possible to figure out how the global open data should be exploited for the surveillance purpose. Integration of the open data with maritime confidential data can improve the efficiency of maritime surveillance and should be considered as a further improvement of the system. Another improvement can be considering a probability for each detected anomaly according to the history of the vessels behavior and the current situation. Moreover, further investigation on the other sources of open data such as social data, which is created and shared through social media platforms, and online videos from the ports activities in the high risk regions, will be useful. The data that are used in ODADS are relatively trusted, but in case of using other open data sources in the maritime surveillance domain for anomaly detection, the quality assurance of the data should be investigated. As well as using knowledge-based systems, taking advantage of data-driven approaches such as machine learning techniques can increase the efficiency

of the maritime surveillance systems. Finally, the next step for improving the maritime surveillance systems after being equipped with the anomaly detection functionality is to predict the future threats or incoming anomalies based on the analysis of the current situation.

References

[1] E. Lefebvre, M. Simard, and C. Helleur. "Multisource information adaptive fuzzy logic correlator for recognized maritime picture". In: *Int'l Conf. on Artificial Intelligence and Applications Symp.* 1. 2001, pp. 229–234.

[2] A. Ponsford, I. A. D'Souza, and T. Kirubarajan. "Surveillance of the 200 nautical mile EEZ using HFSWR in association with a spaced-based AIS interceptor". In: *Conf. on Technologies for Homeland Security*. IEEE. 2009, pp. 87–92.

[3] J. C. Molloy. "The open knowledge foundation: Open data means better science". In: *PLoS Biology* 9.12 (2011), e1001195.

[4] J. Alonso, O. Ambur, M. A. Amutio, O. Azañón, D. Bennett, R. Flagg, D. McAllister, K. Novak, S. Rush, and J. Sheridan. "Improving access to government through better use of the web". In: *World Wide Web Consortium* (2009).

[5] D. Dietrich, J. Gray, T. McNamara, A. Poikola, R. Pollock, J. Tait, and T. Zijlstra. *Open Data Handbook Open Knowledge Foundation Logo*. 2009. URL: `www.opendatahandbook.org/en/`.

[6] S. Dedijer and N. Jéquier. *Intelligence for economic development: An inquiry into the role of the knowledge industry*. Berg Pub Ltd, 1987.

[7] V. Chandola, A. Banerjee, and V. Kumar. "Anomaly detection: A survey". In: *ACM Computing Surveys* 41.3 (2009), p. 15.

[8] J. Roy. "Anomaly detection in the maritime domain". In: *Optics and Photonics in Global Homeland Security IV*. Vol. 6945. Int'l Society for Optics and Photonics. 2008, 69450W.

[9] A. B. Guyard and J. Roy. "Towards case-based reasoning for maritime anomaly detection: A positioning paper". In: *Proc. of The IASTED Int'l Conf. on Intelligent Systems and Control*. Vol. 665. 006. 2009, p. 1.

[10] M. Nilsson, J. Van Laere, T. Ziemke, and J. Edlund. "Extracting rules from expert operators to support situation awareness in maritime surveillance". In: *The 11th Int'l Conf. on Information Fusion*. IEEE. 2008, pp. 1–8.

[11] J. Roy and M. Davenport. "Exploitation of maritime domain ontologies for anomaly detection and threat analysis". In: *Int'l Conf. on Waterside Security*. IEEE. 2010, pp. 1–8.

[12] J. Edlund, M. Grönkvist, A. Lingvall, and E. Sviestins. "Rule-based situation assessment for sea surveillance". In: *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*. Vol. 6242. Int'l Society for Optics and Photonics. 2006, p. 624203.

[13] J. Roy. "Rule-based expert system for maritime anomaly detection". In: *Sensors, and Command, Control, Communications, and Intelligence Technologies for Homeland Security and Homeland Defense IX*. Vol. 7666. Int'l Society for Optics and Photonics. 2010, 76662N.

[14] F. Fooladvandi, C. Brax, P. Gustavsson, and M. Fredin. "Signature-based activity detection based on Bayesian networks acquired from expert knowledge". In: *The 12th Int'l Conf. on Information Fusion*. IEEE. 2009, pp. 436–443.

[15] F. Johansson and G. Falkman. "Detection of vessel anomalies-a bayesian network approach". In: *The Third Int'l Conf. on Intelligent Sensors, Sensor Networks and Information*. IEEE. 2007, pp. 395–400.

[16] R. O. Lane, D. A. Nevell, S. D. Hayward, and T. W. Beaney. "Maritime anomaly detection and threat assessment". In: *The 13th Conf. on Information Fusion*. IEEE. 2010, pp. 1–8.

[17] A. Dahlbom and L. Niklasson. "Trajectory clustering for coastal surveillance". In: *The Tenth Int'l Conf. on Information Fusion*. IEEE. 2007, pp. 1–8.

[18] M. Riveiro, G. Falkman, and T. Ziemke. "Improving maritime anomaly detection and situation awareness through interactive visualization". In: *The 11th Int'l Conf. on Information Fusion*. IEEE. 2008, pp. 1–8.

[19] B. J. Rhodes, N. A. Bomberger, M. Seibert, and A. M. Waxman. "Maritime situation monitoring and awareness using learning mechanisms". In: *Conf. on Military Communications*. IEEE. 2005, pp. 646–652.

[20] R. Laxhammar. "Anomaly detection for sea surveillance". In: *The 11th Int'l Conf. on Information Fusion*. IEEE. 2008, pp. 1–8.

[21] M. Andersson and R. Johansson. "Multiple sensor fusion for effective abnormal behaviour detection in counter-piracy operations". In: *Int'l Conf. on Waterside Security*. IEEE. 2010, pp. 1–7.

[22] B. Ristic, B. La Scala, M. Morelande, and N. Gordon. "Statistical analysis of motion patterns in AIS data: Anomaly detection and motion prediction". In: *The 11th Int'l Conf. on Information Fusion*. IEEE. 2008, pp. 1–7.

[23] C. Carthel, S. Coraluppi, and P. Grignan. "Multisensor tracking and fusion for maritime surveillance". In: *The Tenth Int'l Conf. on Information Fusion*. IEEE. 2007, pp. 1–6.

[24] M. Guerriero, P. Willett, S. Coraluppi, and C. Carthel. "Radar/AIS data fusion and SAR tasking for maritime surveillance". In: *The 11th Int'l Conf. on Information Fusion*. IEEE. 2008, pp. 1–5.

[25] B. J. Rhodes, N. A. Bomberger, M. Seibert, and A. M. Waxman. "SeeCoast: Automated port scene understanding facilitated by normalcy learning". In: *Conf. on Military Communications*. IEEE. 2006, pp. 1–7.

[26] M. Vespe, M. Sciotti, F. Burro, G. Battistello, and S. Sorge. "Maritime multi-sensor data association based on geographic and navigational knowledge". In: *Radar Conf.* IEEE. 2008, pp. 1–6.

[27] S. F. Andler, M. Fredin, P. M. Gustavsson, J. van Laere, M. Nilsson, and P. Svenson. "SMARTracIn: A concept for spoof resistant tracking of vessels and detection of adverse intentions". In: *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense VIII*. Vol. 7305. Int'l Society for Optics and Photonics. 2009, 73050G.

[28] Z. Ding, G. Kannappan, K. Benameur, T. Kirubarajan, and M. Farooq. "Wide area integrated maritime surveillance: An updated architecture with data fusion". In: *Proc. of the Sixth Int'l Conf. on Information Fusion*. Vol. 2. 2003, pp. 1324–1333.

[29] E. Lefebvre and C. Helleur. "Automated association of track information from sensor sources with non-sensor information in the context of maritime surveillance". In: *Proc. of the Seventh Int'l Conf on Information Fusion*. Citeseer. 2004.

[30] J.-P. Mano, J.-P. Georgé, and M.-P. Gleizes. "Adaptive multi-agent system for multi-sensor maritime surveillance". In: *Advances in Practical Applications of Agents and Multiagent Systems*. Springer, 2010, pp. 285–290.

[31] M. Riveiro and G. Falkman. "Interactive visualization of normal behavioral models and expert rules for maritime anomaly detection". In: *The Sixth Int'l Conf. on Computer Graphics, Imaging and Visualization*. IEEE. 2009, pp. 459–466.

[32] J. Van Laere and M. Nilsson. "Evaluation of a workshop to capture knowledge from subject matter experts in maritime surveillance". In: *The 12th Int'l Conf. on Information Fusion*. IEEE. 2009, pp. 171–178.

[33] M. Hall, S. Hall, and T. Tate. "Removing the HCI bottleneck: How the human computer interface (HCI) affects the performance of data fusion systems". In: *Proc. 2000 Meeting of the MSS, Nat. Symp. Sensor and Data Fusion*. 2000, pp. 89–104.

[34] L. L. Constantine and L. A. Lockwood. *Software for use: A practical guide to the models and methods of usage-centered design*. Pearson Education, 1999.

[35] J. Han, J. Pei, and M. Kamber. *Data mining: Concepts and techniques*. Elsevier, 2011.

Appendix

Table A1. A list of some maritime open data sources available on the Internet
Table A2. Data sources that are used in the implementation
Table A3. Total number of vessels in the surveillance area during one week of the system execution
Table A4. The average number of detected anomalies during one week of execution

Table A1. A list of some maritime open data sources available on the Internet

Organization name	Categories	NAR	AR		Provider
			FR	NFR	
European Maritime Safety Agency www.emsa.europa.eu/oil-recovery-vessels/vessel-technical-specifications.html	Maritime safety Prevention of pollution from ships	X			Portugal
ICC Commercial Crime Services www.icc-ccs.org/	Fraud in international trade	X		X	UK
International Association of Classification Societies www.iacs.org.uk/shipdata/default.aspx	Maritime safety Regulation	X		X	UK
International Group of P&I Clubs www.igpandi.org/Home	liability and insurance issues	X	X	UK	
World Shipping Register www.world-register.org/	Ships information Ports information Companies information	X			–
Lloyd's Register Ships In Class www.lrshipsinclass.lrfairplay.com/default.aspx	Ships information		X		UK
International Telecommunication Union www.itu.int/ITU-R/index.asp?category=terrestrial&rlink=mars&lang=en	Ships information Addresses of accounting authorities, administrations which notify information Coasts information MMSI assigned to search and rescue aircraft MMSI assigned to AIS Aids to Navigation	X		X	Switzerland

Continued

Organization name	Categories	NAR	AR		Provider
			FR	NFR	
International Maritime Consultancy Specialists in VTS www.maritime-vts.co.uk/	Vessel traffic services and maritime organization	X			UK
Port Directory www.port-directory.com/	Ports information	X			UK
Equasis www.equasis.org/EquasisWeb/public/HomePage?fs=HomePage	Ships information Companies information		X		Portugal
Q88.COM www.q88.com/Home.aspx?c=1	Questionnaire generator Ships information	X		X	USA
InforMare www.informare.it/indexuk.htm	Shipping information	X			Italy
Paris Mou www.parismou.org/	Port State control	X			Netherlands
Tokyo Mou www.tokyo-mou.org/	Port State control	X			Japan
Australian Government Bureau of Meteorology www.bom.gov.au/	Weather, climate and water	X		X	Australia
Finnish Meteorological Institute en.ilmatieteenlaitos.fi/home	Weather, climate and water	X			Finland
Meteo France france.meteofrance.com/	Weather, climate and water	X		X	France
Earth Science Office NASA weather.msfc.nasa.gov/GOES/	Weather, climate and water	X			–

Continued

Organization name	Categories	NAR	AR		Provider
			FR	NFR	
The Weather Channel www.weather.com/	Weather, climate and water	X			–
Ocean Color WEB NASA oceancolor.gsfc.nasa.gov/	Weather, climate and water	X		X	–
Earth European Space Agency earth.esa.int/ers/eeo4.10075/atsr_med.html	Weather, climate and water		X	X	Italy
Weather BBC www.bbc.co.uk/weather/	Weather, climate and water	X			UK
Sailwx.info www.sailwx.info/	Live marin information	X			USA
Ship.gr www.ship.gr/	Ship brokers information Ship suppliers Companies information	X			–
American Bureau of Shipping www.eagle.org/eagleExternalPortalWEB/appmanager/absEagle/absEagleDesktop?_nfpb=true&_pageLabel=abs_eagle_portal_home_page	Classification Societies	X			USA
Det Norske Veritas www.dnv.com/	Classification Societies	X		X	Norway
Bureau Veritas Groups www.bureauveritas.com/wps/wcm/connect/bv_com/Group/Footer/Home/	Classification Societies	X		X	–
China Classifiaction Societies www.ccs.org.cn/en/index.htm	Classification Societies			X	China

Continued

Organization name	Categories	NAR	AR		Provider
			FR	NFR	
HELLENIC REGISTER OF SHIPPING www.hrs.gr/index.htm	Classification Societies	X			Greece
Nippon Kaiji Kyokai www.classnk.or.jp/hp/en/index.aspx	Classification Societies	X			Japan
Vesseltracker.com www.vesseltracker.com/en/VesselArchive.html	AIS Data Ships information	X	X		Germany
Digital Seas www.digital-seas.com/start.html	Ships information AIS Data	X	X		Germany
MarineTraffic.com www.marinetraffic.com/ais/	Ships information AIS Data	X			Greece
Shipspotting.com www.shipspotting.com/	Ships information AIS Data	X	X		–
International Maritime Organization www5.imo.org/SharePoint/mainframe.asp?topic_id=334&offset	Piracy reports	x			UK
Copenhagen Malmö Port www.cmport.com/	Port Authorities	X			Denmark
Port of Gothenburg www.portgot.se/prod/hamnen/ghab/dalis2b.nsf	Port Authorities	X			Sweden
Swedish Maritime Administration (Sjofartsverket) http://www.sjofartsverket.se/sv/	Pilotage Fairway Service Maritime Traffic Information Icebreaking Hydrograpy	X		X	Sweden

Continued

Organization name	Categories	NAR	AR		Provider
			FR	NFR	
	Maritime and Aeronautical Search and Rescue Seamen's Service				
Ports of Stockholm http://www.stockholmshamnar.se/	Port Authorities	X			Sweden
Port of Norrköping http://www.norrkoping-port.se/	Port Authorities	X			Sweden
Port of Helsinki http://www.portofhelsinki.fi	Port Authorities	X			Finland
Port of Tallinn http://www.ts.ee/	Port Authorities	X			Estonia
Genoa Port Authority www.porto.genova.it/index.php/en	Port Authorities	X			Italy
Port of Klaipeda www.portofklaipeda.lt/en.php	Port Authorities	X			Lithuania
Philippine Ports Authority www.ppa.com.ph/	Port Authorities	X			Philippine
Panama Canal Authority www.pancanal.com/eng/index.html	Port Authorities	X			USA
UK P&I CLUB	liability and insurance issues	X	X		UK

Continued

Organization name	Categories	NAR	AR		Provider
			FR	NFR	
www.ukpandi.com/					
The American Club www.american-club.com/	liability and insurance issues	X	X		USA
Steamship Mutual www.simsl.com/	liability and insurance issues	X	X		Bermuda
SKULD www.skuld.com/	liability and insurance issues	X			Norway
North of England P&I Association www.nepia.com/home/	liability and insurance issues	X			UK
The standard club www.standard-club.com/	liability and insurance issues	X		X	UK
Baltic Ports Organization www.bpoports.com/	Port coordinator	X		X	Denmark

Note. NAR = no authorization required; AR= authorization required; FR = free registration; NFR = non-free registration; Dashes indicate undisclosed information.

Table A2. Data sources that are used in the implementation

Website name	Data type
Marinetraffic.com	Real time information based on AIS systems[1]
Swedish Maritime Administration (Sjöfartsverket)	Stockholm pilotage area[2]
Ports of Stockholm, Kapellskär and Nynäshamn	Vessels in port and expected arrival[3]
Port of Norrköping	Vessels in port[4]
	Expected vessels arrival[5]
Port of Helsinki	Cargo vessels in port[6]
	Expected cargo vessels arrival[7]
	Expected passenger vessels departure[8]
	Expected passenger vessels arrival[9]
	Passenger vessels have visited the port before[10]
Port of Tallinn	Vessels in port[11]
	Expected cargo vessels arrival[12]
	Expected passenger vessels arrival[13]
	Expected passenger vessels departure[14]

[1] www.marinetraffic.com/ais/
[2] www.sjofartsverket.se/sv/Infrastruktur-amp-Sjotrafik/Lotsning/Lotsinfo/
[3] stockholmshamnar.se/en/Karta/Vessel-calls/
[4] www.norrkoping-port.se/anlop.php?page=snabb_fih&link=110|111
[5] www.norrkoping-port.se/anlop.php?page=snabb_fih_ank&link=110|111
[6] www.portofhelsinki.fi/cargo_traffic/vessels_in_ports
[7] www.portofhelsinki.fi/cargo_traffic/arrival_ships
[8] www.portofhelsinki.fi/passengers/departure_times_and_terminals
[9] www.portofhelsinki.fi/passengers/arrival_times_and_terminals
[10] www.portofhelsinki.fi/passengers/cruise_ships_that_have_visited_the_port
[11] www.ts.ee/?op=ships_in_port&lang=eng
[12] www.ts.ee/?op=cargo_ships_arrivals&lang=eng
[13] www.ts.ee/?op=passenger_ship_arrivals&lang=eng
[14] www.ts.ee/?op=passenger_ship_departures&lang=eng

Table A3. Total number of vessels in the surveillance area during one week of the system execution

	Count (Avg)[1]
Total number of vessels	673.29
Cargo, Tanker, Passenger and Ferry vessels	366.29
Cargo, Tanker, Passenger and Ferry vessels that are originated from or targeted to the specified ports	141.71

[1] The daily average count

Table A4. The average number of detected anomalies during one week of execution

Anomaly	Count (Avg)[1]
ARRIVAL_TIME_MISMATCHED	23.86
VESSEL_NOT_INFORMED_PORT	14.71
VESSEL_NOT_LEFT_PORT	8.29
UNUSUAL_TRIP_PATTERN	3.71
VESSEL_NOT_USED_PILOT	2.00
UNUSUAL_TRIP_PATTERN_AND_VESSEL_ARRIVAL_TIME_MISMATCHED	1.71
VESSEL_MOORED_IN_PORT	1.29
UNUSUAL_TRIP_PATTERN_AND_VESSEL_NOT_INFORMED_PORT	0.57
VESSEL_ENTERED_PORT_WITHOUT_NOTICE	0.29
VESSEL_NOT_ENTERED_PORT	0.29
UNDER_SURVEILLANCE_VESSEL	0.29
VESSEL_ARRIVAL_TIME_MISMATCHED_AND_VESSEL_ NOT_USED_PILOT	0.29
VESSEL_ORDERED_PILOT_AND_NOT_INFORMED_PORT	0.14
UNUSUAL_TRIP_PATTERN_AND_VESSEL_NOT_USED_PILOT	0.14
VESSEL_ORDERED_PILOT_AND_NOT_INFORMED_PORT_ AND_VESSEL_NOT_USED_PILOT	0.14
UNUSUAL_TRIP_PATTERN_AND_VESSEL_ORDERED_PILOT_ AND_NOT_INFORMED_PORT	0.00
UNUSUAL_TRIP_PATTERN_AND_VESSEL_NOT_USED_PILOT_ AND_VESSEL_ARRIVAL_TIME_MISMATCHED	0.00
VESSEL_ENTERED_PORT_WITHOUT_NOTICE_AND_NOT_LEFT_ PORT_ON_TIME	0.00
VESSEL_NOT_ENTERED_PORT_AND_NOT_USED_PILOT	0.00
UNUSUAL_TRIP_PATTERN_AND_VESSEL_NOT_ENTERED_PORT	0.00
UNUSUAL_TRIP_PATTERN_AND_VESSEL_NOT_ENTERED_PORT_ AND_VESSEL_NOT_USED_PILOT	0.00
UNUSUAL_TRIP_PATTERN_AND_VESSEL_ORDERED_PILOT_AND_ NOT_INFORMED_PORT_AND_VESSEL_NOT_USED_PILOT	0.00

[1]The daily average count

9

Trend Analysis to Automatically Identify Heat Program Changes

Shahrooz Abghari, Eva Garcia-Martina, Christian Johansson, Niklas Lavesson, Håkan Grahn
In: Journal of Energy Procedia 116 (2017): pp. 407-415.
Also Publised In: The 15th International Symposium on District Heating and Cooling, 2016, Seoul, Korea.

Abstract

The aim of this study is to improve the monitoring and controlling of heating systems located at customer buildings through the use of a decision support system. To achieve this, the proposed system applies a two-step classifier to detect manual changes of the temperature of the heating system. We apply data from the Swedish company NODA, active in energy optimization and services for energy efficiency, to train and test the suggested system. The decision support system is evaluated through an experiment and the results are validated by experts at NODA. The results show that the decision support system can detect changes within three days after their occurrence and only by considering daily average measurements.

9.1 Introduction

In the district heating (DH) domain, operators address several conflicting goals, such as satisfying customer demand while minimizing production and distribution costs. To achieve this, one solution is to equip each customer building with a smart system. Such a system should continuously monitor heat usage, predict future demand, exchange information with operators, and perform demand-side management. Moreover, the system needs to automatically learn the energy usage of the building and adopt its behavior

accordingly. NODA Intelligent Systems AB[1] , an active company in the DH domain, is developing and providing retrofit smart systems to maximize energy efficiency in buildings. These systems consist of controlling hardware together with a range of sensors, which are added on top of the existing control system.

Self-learning and adaptation are two important features of any smart system. However, these two features make the system sensitive to manual changes in the heating system, forcing the system to re-learn its characteristics. Most commonly this relates to applying changes in the temperature program of the controller e.g. by the owner's building. These changes can lead to use more energy and to add extra charges in the case of increasing the temperature of the system.

Although retrofit solutions such as NODA's smart system can decrease the cost of replacement of the existing control system, their functionality can be affected by the limitation of these existing controllers. Due to this reason, NODA's smart system is unable to detect manual changes online. Hence, NODA's operators need to spend significant efforts to detect the manual changes by analyzing the received information from each building controller. To make this process more efficient, a decision support (DS) system can be used to assist operators. DS systems are computer-based information systems, which aim to facilitate and support the decision-making processes [1]. The major components of DS systems are: 1) the user-interface, 2) the models and main logic, 3) the database, and 4) the DS functionalities and architecture. DS systems are categorized based on their functionalities into: data-driven, knowledge-driven, model-driven, document-driven and communication-driven DS systems [1]. Among these different types, data-driven systems can provide an online support for decision making through applying machine learning (ML) and statistical techniques to analyze large collections of data. Machine learning is a branch of artificial intelligence, which includes the study of algorithms that can learn and improve their knowledge by building models from input data to perform specific tasks. Most common tasks in ML, such as classification and regression modeling, are solved with supervised learning methods. Supervised learning uses labeled data to train models [2]. Suppose we are given data in the form of $(\overrightarrow{x_1}, y_1), (\overrightarrow{x_2}, y_2), ..., (\overrightarrow{x_n}, y_n)$. In each pair or instance $\overrightarrow{x_i}$ (input) denotes a vector, which consists of feature values such as indoor and outdoor tempera-

[1] www.noda.se/en/main

ture, and y_i (output) indicates a label or outcome of the target attribute. The aim is to train a model to predict the label of the target attribute (y_i) of each new instance, e.g. predicting the secondary supply temperature based on the indoor and outdoor temperature. The target attribute in regression modeling is numeric and in classification modeling it is categorical.

In this paper, we propose a data-driven decision support system that uses ML techniques to detect manual changes by predicting the secondary supply temperature based on the outdoor temperature and analyzing the energy consumption of each building. The aim of such a system is to provide complementary decision support for NODA's operators to detect manual changes easily and efficiently. The proposed DS system uses a two-step classifier, a combination of k-means and support vector regression (SVR), to detect manual changes within three days after their occurrence by considering daily average measurements.

9.2 Background and Related Work

A district heating system (DHS) is a centralized system with the aim of producing space heating and hot tap water for consumers based on their demand at a limited geographic area. A DH system consists of three main parts: production units, distribution network, and consumers. The heated water supplied in a production unit circulates through the distribution network and will be available to consumers. The main aim of a DHS is to minimize the cost and pollution by considering consumers' demand and producing just the necessary amount of heat. Hence, being able to predict the heat demand can assist production units to plan better. However, modeling the heat demand forecasting is a challenging task, since water does not move fast. In some situations, the distribution of heated water can take several hours. Moreover, there are a number of factors that affect the forecast accuracy and need to be considered before any plan for production units can be constructed. Some of these factors include [3, 4]:

1. Weather condition, mainly the outdoor temperature
2. Social behavior of the consumers
3. Irregular days such as holidays

4. Periodic changes in conditions of heat demand such as seasonal, weekly and day-night

Fumo [5] pointed out in his review two commonly used techniques for energy demand estimation, namely; forward (classical) and data-driven (inverse) techniques. The first approach describes the behavior of systems by applying mathematical equations and known inputs to predict the outputs. In contrast, data-driven techniques use ML methods to learn the system's behavior by building a model with training data in order to make predictions.

Dotzauer [4] introduced a very simple model for forecasting heat demand based on outdoor temperature and social behavior. He showed that the predictions of his simple model were comparable with complicated models such as autoregressive moving average model (ARMA). The author concluded that better predictions can be achieved by improving the weather forecasts instead of developing complicated heat demand forecasting models.

In general, different ML methods and techniques have been used to predict the heat demand. Some of the most popular prediction models are autoregressive moving average (ARMA) [6], support vector regression (SVR) [7, 8], multiple linear regression (MLR) [9] and artificial neural network (ANN) [10, 11]. In [8], the authors compared four supervised ML methods for building short-term forecasting models. The models are used to predict heat demand for multi-family apartment buildings with different horizon values between 1 to 24 hours ahead. The authors concluded that SVR achieves the best performance followed by MLR in comparison to feed forwards neural network (FFNN), and regression trees methods. Recently, Provatas et al. [12], proposed the usage of on-line ML algorithms in combination with decision tree-based ML algorithms for heat load forecasting in a DH system. The authors investigated the impact of two different approaches for heat load aggregation. The results of the study showed that the proposed algorithm has a good prediction result. In another study [13], the authors showed the application of a context vector (CV) based approach for forecasting energy consumption of single family houses. The proposed method is compared with linear regression, K-nearest neighbors (KNN) and SVR methods. The results of the experiment showed that CV performed better in most cases followed by KNN and SVR. The authors concluded the proposed solution can help DH companies to improve their schedule and reduce operational costs.

There are a number of studies that focused on the application of DS systems in domains such as DH and mainly related to advanced energy management [14–19]. In these studies, the main focus is on forecasting and optimization methods that facilitate and support the decision-making processes to increase the energy management quality and bring considerable savings. Furthermore, there are some other works that focused on DH network design [20, 21]. Bordin et al. [20] presented a mathematical model to support DH system network planning by selecting an optimal set of new users to be connected to a thermal network that maximizes revenues and minimizes infrastructure and operational costs.

In summary, the main focus of the studies that have been done in the context of heat demand forecasting in the DH domain was related to using weather forecast data and mainly the outdoor temperature. In contrast, the aim of the proposed solution in this study is twofold: 1) to provide decision support for operators to detect manual changes efficiently, and 2) to decrease the energy consumption cost and control heat demand by identifying these changes and resolving them at each building.

9.3 Detection of Changes in Trends by Using Regression Methods

In DH, operators try to address several conflicting goals, such as satisfying customer demand while minimizing production and distribution costs. One way to solve this is to use demand side management and data analytics in the customer substations. This can be achieved by a system that continuously predicts the future heat demand, exchanges information with the operator and performs demand side management when the need arises. Such systems can be implemented both in the existing heating controllers as well as in retrofit solutions. One such retrofit system is developed by NODA and it has been used within this study. To make the system efficient, its behavior has to be as automated and self-learning as possible. However, this also makes the system sensitive to manual changes (i.e. changing the temperature) in the heating system, since such changes forces the system to re-learn the characteristics of the heating system.

In order to assist operators to detect these manual changes more efficiently a DS system is implanted to provide decision support for operators. The

proposed DS system uses a two-step classifier, k-means and SVR, to detect manual changes. To achieve this goal and to avoid generating false alarms in confront with noisy data, changes should be monitored for some days. Hence, in this study only those deviations that last for at least 3 consecutive days would be marked as manual changes. k-means, which is the most well-known algorithm for classification task, is used to identify the operational status of the heating system (on or off) by partitioning the consumed energy at each building.

The main reason to perform this task is to decrease the effect of outliers when the heating system is not operating. SVR has been used for both electricity and heat demand forecasting and has been found to be very efficient and accurate [8, 22]. Therefore, SVR is chosen to predict secondary supply temperature based on outdoor temperature and consumed energy for each building. By considering the status of the system and the predicted value of the secondary supply temperature, the DS system can identify manual changes as follows:

IF the absolute difference (actual – predicted) is greater than the threshold FOR 3 consecutive days THEN changes have occurred during these days.

The warning threshold determines the sensitivity of the system to change. This threshold, set to 4.6 °C, was determined empirically after performing some preliminary tests and checking the results with the subject matter experts. Figure 9.1 summarizes the process of automatically identifying the manual changes for each building by the proposed DS system.

9.3.1 Algorithms

The k-means algorithm belongs to the group of distance-based clustering methods. It is the best known greedy algorithm for partitioning data into k clusters. This popularity is mainly related to k-means' simplicity, efficiency, and applied success in partitioning and pattern recognition tasks [2, 23]. It works by reducing the total sum of the squared error over all k clusters. k-means iterates by generating partitions and assigning data to the closest cluster and computing the centroid from a partition until no further improvement can be achieved [23].

The support vector machine (SVM) algorithm is based on statistical learning theory. SVM is a state-of-the-art algorithm, which belongs to a

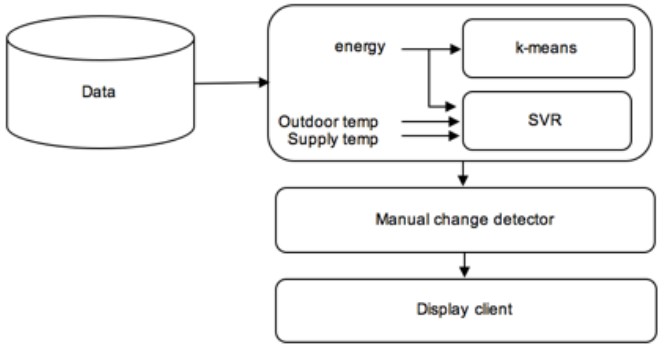


Figure 9.1: The process of automatically identifying the manual changes for each building by the DS system.

group of supervised learning methods that can solve different ML tasks such as classification, pattern recognition and regression [24]. An extended version of SVM for regression tasks is called support vector regression. SVR uses the training data to find the regression line that best fits the data. Using an epsilon-intensive loss function, SVR produces a decision boundary, a subset of training data which is called support vectors (SVs), in order to determine a tube with radius ε fitted to the data. In other words, epsilon defines how well the regression line fits the data by ignoring errors as long as they are less than ε.

9.4 Research Method

9.4.1 Data Collection

The data used in this study consists of daily average measurements from 9 buildings equipped with the NODA controller. The buildings are located in Karlshamn in south Sweden. The collected data was obtained on the period between April 2014 and March 2016. This yields 730 instances per building (one instance per day). However, since data collection instruments, such as sensors, might be faulty, or since data transmission errors can occur [25], some of the measurements were incomplete. Therefore, after performing the data cleaning process the number of instances decreased to approximately 630 per building. Table 9.1 summarizes the information and the way the

data is split to train and test set for each building.

Table 9.1: Summary of the data collection for each building

Building ID	Data *(no. of instances)*		
	Train set (Apr 2014 - Mar 2015)	Test set (Apr 2015 - Mar 2016)	Total
A	251	249	500
B	349	365	714
C	357	365	722
D	357	365	722
E	347	347	694
F	270	365	635
G	251	249	500
H	357	332	689
I	362	345	707

Figure 9.2 shows the daily average of the secondary supply temperature of building D with respect to the outdoor temperature for the year 2015 (365 instances). The plot shows that the secondary supply temperature has a strong correlation with the outdoor temperature.

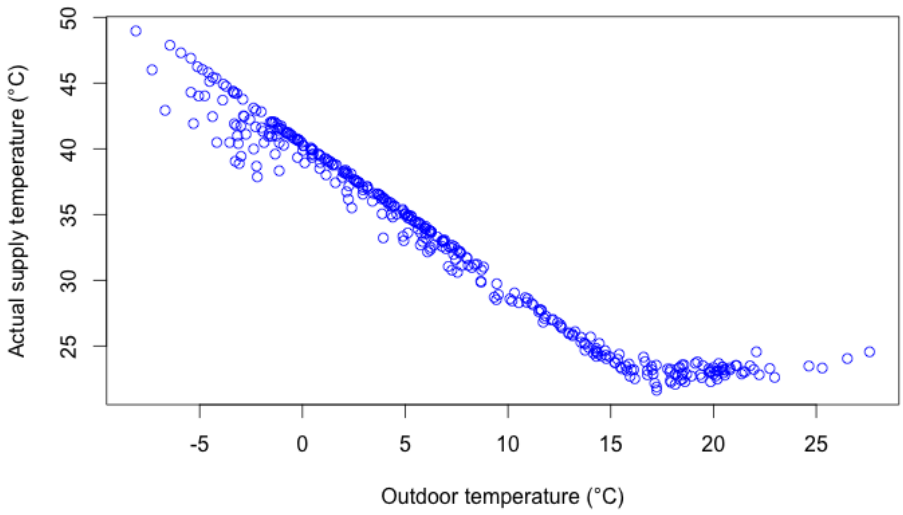


Figure 9.2: Daily average secondary supply temperature of building D with respect to the daily average outdoor temperature for the year 2015 (365 instances).

We used R and RWeka package to conduct the experiment. R is a language and a free software environment for statistical computing with data [26]. R is widely used for visualization and statistical tasks such as linear and non-linear modelling, regression analysis, and statistical tests. RWeka is an R interface to WEKA (Waikato environment for knowledge analysis) [27]. WEKA [28] is a well-known machine learning and data mining workbench written in Java. It contains a wide range of algorithms for different ML tasks such as classification, regression, and clustering. We used RWeka's k-means and SVR implementation with their default parameters.

9.4.1.1 Experimental Design

To detect manual changes in the heating system for each building, the implemented DS system uses a two-step classifier. 10-fold cross validation is used on data from April 2014 to March 2015 to build the model for each building. m-fold cross validation is a standard procedure for a model evaluation in ML. The main idea is to randomly split the dataset into m equal subsets. The model is trained and tested m times. Each time one of the m subsets is used as a test set and the other $m-1$ subsets are form the training set. The overall performance of the model is computed as the average error across all m runs [29]. The train set is preprocessed and cleaned to make sure that the DS system only learns the normal behavior of the heating system. Additionally, the quality of the model is tested with the data from April 2015 to March 2016.

The performance of the system is evaluated in two ways:

1. using mean absolute error (MAE) as a performance measure to evaluate the accuracy of SVR in terms of predicting the secondary supply temperature.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|actual_i - predicted_i| \tag{9.1}$$

   In equation (9.1), the actual refers to the measured secondary supply temperature by the controller system, predicted refers to the estimated secondary supply temperature by the proposed DS system, and n is the total number of predicted instances.

2. validating the detected changes by subject matter experts at NODA. In this case the accuracy of the system is calculated based on the

number of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) alarms in equation (9.2). The TPs and TNs are correct classifications. A false positive happens when the result is classified incorrectly as a detected change while it is actually not a change. A false negative occurs when an actual change in the system is not detected [25].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9.2}$$

9.5 Results

The performance of the proposed DS system is evaluated by using the test data (April 2015 – March 2016) and 10-fold cross validation for each building. Furthermore, the identified changes at each building is validated with NODA's experts. Table 9.2 summarizes the performance of the SVR together with the number of manual detected changes at each building. The

Table 9.2: Mean absolute error and standard deviation for SVR and detected changes for each individual building

Building ID	MAE	Detected changes			
		TP	TN	FP	FN
A	1.64 (±0.016)	26	223	-	-
B	1.70 (±0.008)	-	358	7	-
C	2.40 (±0.006)	8	357	-	-
D	0.72 (±0.002)	-	365	-	-
E	1.07 (±0.003)	-	320	27	-
F	0.73 (±0.004)	-	365	-	-
G	1.64 (±0.016)	26	223	-	-
H	0.49 (±0.002)	-	332	-	-
I	1.02 (±0.004)	-	345	-	-
Total	-	60	2,888	34	0

Note. MAE = mean absolute error, standard deviation appears within the parentheses.

results show that the DS system detected, in total 60 changes correctly in 3 out of 9 buildings. These changes either related to manual changes or hardware failures. This value represents the number of TP alarms. The majority of the results belonged to the TN category with the value of 2,888. The false positive alarms occurred in 2 buildings and in total contain 34

changes. The main reason for these detected changes are related to a sudden drop in the outdoor temperature, and the fact that the system was not trained for such a situation. No false negative is detected during the experiment. By considering these values and using the equation (9.2) the accuracy of the system can be computed as follow: $(60 + 2,888)/(60 + 2,888 + 34) = 0.98$.

Figures 9.3 and 9.4 depict the outcome of the system for two different buildings. Figure 3 shows the detected manual changes occurred during 14^{th} until 23^{rd} of January 2016 at the C building. These manual changes are related to the modification of the temperature of the heating system. Figure 4 is related to the D building. This building has no changes, which can be seen since the actual and predicted secondary supply temperature are closely following each other.

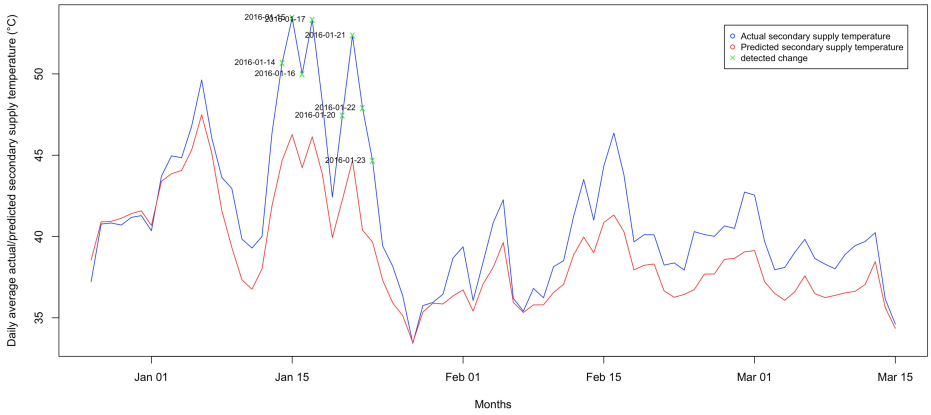


Figure 9.3: Identified manual changes during January 2016 at the C building. The actual secondary supply temperature is showed in blue against the predicted secondary supply temperature in red. The green crosses identify the detected manual changes by the DS system.

9.6 Discussion

The experimental results show that the proposed DS system with a two-step classifier is able to detect manual changes within three days after their occurrence. The accuracy of the system is evaluated by the experts from NODA. The results of the evaluation show that the system has a solid

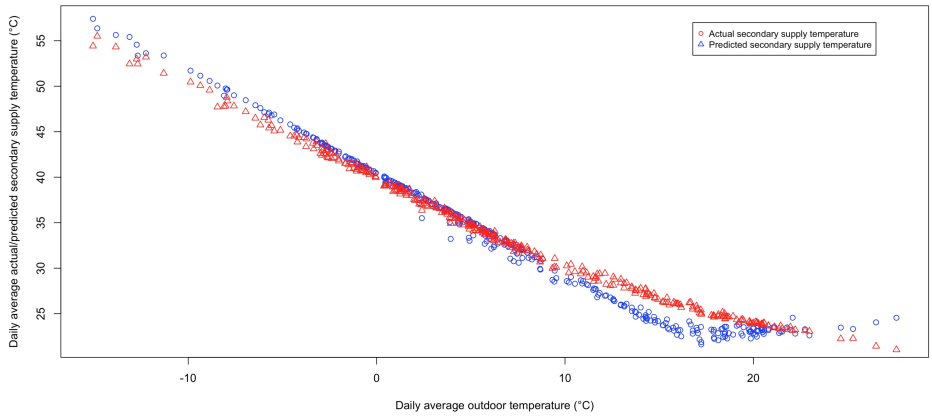


Figure 9.4: The actual and predicted secondary supply temperature related to building D. This building has no changes during April 2015 – March 2016.

detection ability with an accuracy of 98%. In general, the important aspect of such system is its ability to detect changes correctly and does not miss any changes.

To decrease the false alarms (both FP and FN) in the detection task, the main solution is to train the system with the data that represents the normal behavior of the heating system. Moreover, only those deviations that last at least three days are classified as manual changes. In addition, considering an adjustable threshold instead of a fix one can decrease the number of false alarms. Though the false positive alarms can be quickly determined and dismissed by experts, considerable number of false alarms can be disturbing.

The proposed DS system is generalizable to similar applications such as detection of change in energy demand or detection of faulty equipment based on abnormal behavior of the heating system.

9.7 Conclusion

We propose a decision support system for operators in the district heating domain. Currently, the proposed system is applied to detect manual changes in the heating system at the building level. The decision support system

uses a two-step classifier, k-means and support vector regression, to identify manual changes within three days after their occurrence and by considering daily average measurements. The performance of the system is evaluated with the real data related to 9 buildings in Sweden. The validity of the results was investigated by the experts at the NODA Intelligent Systems AB. The validation of the results showed that the majority detected changes by the system were true alarms.

Since each building has special characteristics, e.g. its geographical location, used construction materials, and the social behavior of its tenants, having a fixed threshold for all buildings is impractical. Hence, in the future, it is important to investigate how to automatically set the threshold value for each building. Moreover, it is more convenient that operators can have interaction with the DS system by providing feedbacks. Thus, the performance of the system can improve through time.

References

[1] D. J. Power. *Decision support systems: Concepts and resources for managers*. Greenwood Publishing Group, 2002.

[2] P. Flach. *Machine learning: The art and science of algorithms that make sense of data*. Cambridge University Press, 2012.

[3] N. Eriksson. *Predicting demand in districtheating systems: A neural network approach*. 2012.

[4] E. Dotzauer. "Simple model for prediction of loads in district-heating systems". In: *Applied Energy* 73.3-4 (2002), pp. 277–284.

[5] N. Fumo. "A review on the basics of building energy estimation". In: *Renewable and Sustainable Energy Reviews* 31 (2014), pp. 53–60.

[6] H. Wiklund. "Short term forecasting on the heat load in a DH-system". In: *Fernwärme Int'l* 20.5-6 (1991), pp. 286–294.

[7] L. Wu, G. Kaiser, D. Solomon, R. Winter, A. Boulanger, and R. Anderson. "Improving efficiency and reliability of building systems using machine learning and automated online evaluation". In: *The 11th Conf. on Systems, Applications and Technology*. IEEE. 2012, pp. 1–6.

[8] S. Idowu, S. Saguna, C. Åhlund, and O. Schelén. "Forecasting heat load for smart district heating systems: A machine learning approach". In: *Int'l Conf. on Smart Grid Communications*. IEEE. 2014, pp. 554–559.

[9] T. Catalina, V. Iordache, and B. Caracaleanu. "Multiple regression model for fast prediction of the heating energy demand". In: *Energy and Buildings* 57 (2013), pp. 302–312.

[10] K. Kato, M. Sakawa, K. Ishimaru, S. Ushiro, and T. Shibano. "Heat load prediction through recurrent neural network in district heating and cooling systems". In: *Int'l Conf. on Systems, Man and Cybernetics*. IEEE. 2008, pp. 1401–1406.

[11] M. Sakawa, K. Kato, and S. Ushiro. "Cooling load prediction in a district heating and cooling system through simplified robust filter and multilayered neural network". In: *Applied Artificial Intelligence* 15.7 (2001), pp. 633–643.

[12] S. Provatas. *An online machine learning algorithm for heat load forecasting in district heating systems*. 2014.

[13] S. Rongali, A. R. Choudhury, V. Chandan, and V. Arya. "A context vector regression based approach for demand forecasting in district heating networks". In: *Int'l Conf. on Innovative Smart Grid Technologies Asia*. IEEE. 2015, pp. 1–6.

[14] K. Mařík, Z. Schindler, and P. Stluka. "Decision support tools for advanced energy management". In: *Energy* 33.6 (2008), pp. 858–873.

[15] D. Chinese and A. Meneghetti. "Optimisation models for decision support in the development of biomass-based industrial district-heating networks in Italy". In: *Applied Energy* 82.3 (2005), pp. 228–254.

[16] P. Bardouille and J. Koubsky. "Incorporating sustainable development considerations into energy sector decision-making: Malmö Flintränen district heating facility case study". In: *Energy Policy* 28.10 (2000), pp. 689–711.

[17] S. N. Petrovic and K. B. Karlsson. "Danish heat atlas as a support tool for energy system models". In: *Energy Conversion and Management* 87 (2014), pp. 1063–1076.

[18] A. Meneghetti and G. Nardin. “Enabling industrial symbiosis by a facilities management optimization approach”. In: *J. of Cleaner Production* 35 (2012), pp. 263–273.

[19] E. Brembilla and A. Sciomachen. *Design and verification of a large size district heating network by a DSS*. 1990.

[20] C. Bordin, A. Gordini, and D. Vigo. “An optimization approach for district heating strategic network design”. In: *European J. of Operational Research* 252.1 (2016), pp. 296–307.

[21] A. Sciomachen and R. Sozzi. “The algorithmic structure of a decision support system for a design of a district heating network”. In: *Computers & Operations Research* 17.2 (1990), pp. 221–230.

[22] B.-J. Chen, M.-W. Chang, et al. “Load forecasting using support vector machines: A study on EUNITE competition 2001”. In: *IEEE Transactions on Power Systems* 19.4 (2004), pp. 1821–1830.

[23] A. K. Jain. “Data clustering: 50 years beyond K-means”. In: *Pattern Recognition Letters* 31.8 (2010), pp. 651–666.

[24] V. Vapnik. *Statistical learning theory. 1998*. Wiley, New York, 1998.

[25] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[26] RDevelopment CORE TEAM, R and others. *R: A language and environment for statistical computing*. 2008.

[27] K. Hornik, C. Buchta, and A. Zeileis. “Open-source machine learning: R meets Weka”. In: *Computational Statistics* 24.2 (2009), pp. 225–232.

[28] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. “The WEKA data mining software: An update”. In: *ACM SIGKDD Explorations Newsletter* 11.1 (2009), pp. 10–18.

[29] R. Kohavi et al. “A study of cross-validation and bootstrap for accuracy estimation and model selection”. In: *IJCAI*. Vol. 14. 2. Montreal, Canada. 1995, pp. 1137–1145.

10

Outlier Detection for Video Session Data Using Sequential Pattern Mining

Shahrooz Abghari, Veselka Boeva, Niklas Lavesson, Håkan Grahn, Jörgen Gustafsson, Junaid Shaikh
In: Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining: Workshop On Outlier Detection De-constructed, 2018, London, UK.

Abstract

The growth of Internet video and over-the-top transmission techniques has enabled online video service providers to deliver high quality video content to viewers. To maintain and improve the quality of experience, video providers need to detect unexpected issues that can highly affect the viewers' experience. This requires analyzing massive amounts of video session data in order to find unexpected sequences of events. In this paper we combine sequential pattern mining and clustering to discover such event sequences. The proposed approach applies sequential pattern mining to find frequent patterns by considering contextual and collective outliers. In order to distinguish between the normal and abnormal behavior of the system, we initially identify the most frequent patterns. Then a clustering algorithm is applied on the most frequent patterns. The generated clustering model together with Silhouette Index are used for further analysis of less frequent patterns and detection of potential outliers. Our results show that the proposed approach can detect outliers at the system level.

10.1 Introduction

The Internet has transformed almost every aspect of human society by enabling a wide range of applications and services such as online video streaming. Subscribers of such services spend a substantial amount of time

online to watch movies and TV shows. This has required online video service providers (OVSPs) to continuously improve their services and equipment to satisfy subscribers' high expectation. According to a study performed by Krishnan and Sitaraman [1], a 2-second delay in starting an online video program causes the viewers to start abandoning the video. For each extra second delay beyond that the viewers' drop-off rate will be increased by 5.8%. Thus, in order for OVSPs to address subscribers' needs it is important to monitor, detect, and resolve any issues or anomalies that can significantly affect the viewers when watching requested video programs. Analyzing massive amounts of video sessions for identifying such abnormal behaviors is like finding a needle in a haystack.

In this study, we use sequential pattern mining in order to analyze video data sequences from an over-the-top video service (a delivery paradigm that uses Internet to deliver video). The video session data has temporal order and contains detailed information regarding which video is requested, what type of device (mobile phone, PC, etc.) is used for watching the video, and the list of occurrences of all event types. The initial assumption with using sequential pattern mining is that frequent patterns can be considered as normal system behavior, while the others can be potential outliers. By applying a clustering method, most frequent patterns can be grouped based on their similarities. Finally, non-most frequent patterns can be evaluated by the created model and their goodness-of-fit identified by applying an internal cluster validation measure such as Silhouette Index [2].

The proposed approach is able to detect outliers by analyzing video event sequences and finding specific patterns that do not commonly occur. Investigating these unexpected patterns can assist online video service providers to identify, diagnose, and resolve possible system level issues. To the best of our knowledge, this is the first study that combines sequential pattern mining and clustering analysis for detecting outliers in online video streaming.

10.2 Background

10.2.1 Frequent Pattern Mining

The application of frequent itemset mining for market-basket analysis was first introduced by Agrawal et al. in 1993 [3]. The aim of such analysis is to reveal the customers' shopping habits and to find out which sets of

products are frequently bought together. The frequent itemset mining can be formulated as follows: let $\mathcal{I} = \{i_1, i_2, ..., i_n\}$ be a set of all items and $\mathcal{T} = \{t_1, t_2, ..., t_j, ..., t_m\}$ a transaction database, where t_j is a set of items that has been bought by a customer ($t_j \subseteq \mathcal{I}$). The aim is to find those sets of items that occur frequently in most of the shopping baskets considering s, the user-specified *support threshold*.

The *support* for a *k-itemset* X, which consists of k items from $\mathcal{I}$, is the number of transactions that contain X as a subset, i.e., ${}_{ST}(X) = |\{t_j | X \subseteq t_j \wedge t_j \in \mathcal{T}\}|$. Note that the support of X can also be defined as the *relative support* which is the ratio of the number of transactions containing X to the total number of transactions in the database $\mathcal{T}$, i.e., ${}_{RelST}(X) = \frac{{}_{ST}(X)}{|\mathcal{T}|}$, such X is frequent if and only if its support is equal or greater than s.

Originally in frequent itemset mining, the order of items in the itemsets is unimportant. Looking at the market-basket analysis, the goal is to find frequent sets of items that are bought together. However, there are some situations in which the order of items inside the itemset is important such as sequence databases. A sequence database consists of ordered sequences of items listed with or without a concrete notion of time [4]. Sequential pattern mining, the problem of finding interesting frequent ordered patterns, was first introduced in 1995 [5].

Let $\mathcal{I} = \{i_1, i_2, ..., i_m\}$ be a set of all items. A sequence α defined as $\langle a_1, a_2, ..., a_j, ..., a_m \rangle$, where a_j is an itemset. Each itemset a_j represents a set of items that happened at the same time. A sequence $\alpha = \langle a_1, a_2, ..., a_m \rangle$ is a subsequence of $\beta = \langle b_1, b_2, ..., b_n \rangle$ if and only if there exist integers $1 \leq k_1 < k_2 < ... < k_m \leq n$ and $a_1 \subseteq b_{k_1}, a_2 \subseteq b_{k_2}, ..., a_m \subseteq b_{k_m}$ [5]. Given a sequence database $\mathcal{T} = \{s_1, s_2, ..., s_n\}$, the support for α is the number of sequences in $\mathcal{T}$ that contain α as a subsequence. Consequently, α is a frequent sequential pattern if its support is equal or greater than user-specified support threshold.

Mining frequent patterns in a large database can lead to generating a huge number of patterns that satisfy the user-specified support threshold. This is due to the fact that if a pattern is frequent, its sub-patterns are also frequent. To mitigate this problem, *closed* and *maximal* frequent pattern mining has been proposed [4]. A frequent pattern α is called:

1. a closed frequent pattern in the database $\mathcal{T}$ if and only if none of its

super-patterns have the same support as α,

2. a maximal frequent pattern in the database $\mathcal{T}$ if and only if none of its super-patterns is frequent [4, 6].

10.2.2 Sequential Pattern Mining Algorithms

Since the introduction of frequent itemset mining and the Apriori algorithm [3], several extensions of this algorithm were developed for both frequent itemset mining and sequential pattern mining. In general, there are two main categories of algorithms suitable for frequent pattern mining: 1) *Apriori-based algorithms* and 2) *Pattern-growth algorithms*. Additionally, from a frequent pattern mining point of view, a sequence database can represent the data either in a *horizontal data format* or *vertical data format* [7]. Therefore, based on these two data formats Apriori-based algorithms can expand to *horizontal data format algorithms* such as AprioriAll [5], and GSP [8] and *vertical data format algorithms* such as SPADE [9], and SPAM [10]. Apriori-based algorithms generate large sets of candidates and repeatedly scan the database for mining sequential patterns which require a lot of memory [11]. To solve this problem, pattern-growth approach as an extension of FP-growth algorithm [11] for frequent itemset mining without candidate generation was proposed. Pattern-growth algorithms such as FreeSpan [12], and PrefixSpan [13] work in a divide-and-conquer fashion and repeatedly divide the database into a set of smaller *projected databases* and mine them recursively.

The most popular pattern-growth algorithm is PrefixSpan. Given a sequence database, $\mathcal{T}$, and a user-specified threshold, *min_sup*, PrefixSpan applies a prefix-projection method to mine sequential patterns in $\mathcal{T}$ through 1) scanning the database once to find all frequent items with a length one, 2) dividing search space into a number of subsets according to the extracted frequent items in the previous step, and 3) constructing projected databases that represent each subset of sequential patterns and mining them recursively. This way, only local frequent sequences will be explored to create sequential patterns in each projected database [7, 13].

In 2004, Pei et al. [14] showed that PrefixSpan has the best overall performance compared to GSP and SPADE, and FreeSpan. Therefore, in this study we choose to use PrefixSpan for extracting sequential patterns in video data sessions.

10.3 Related Work

Barbará et al. [15] proposed an intrusion detection system that applies a frequent itemset technique to discover sets of items that are available in most data chunks. Using a clustering algorithm, these items that are considered as attack-free traffic, are divided into different groups based on their similarities. After creating the clusters, an outlier detection technique is applied to all the data points checking each instance against the set of clusters. Instances that do not belong to any clusters are presumed to be attacks. Recently, Rossi et al. [16] proposed an anomaly detection system for the smart grid domain similar to one considered in [15]. The method proposed by Rossi et al. uses frequent itemset mining on different event types collected from smart meters to separate normal and potential anomalous data points. For further evaluation, a clustering technique with Silhouette Index analysis is applied to detect anomalies.

Hoque et al. [17] developed an anomaly detection system for monitoring daily in-home activities of elderly people called *Holmes*. The proposed system learns a resident's normal behavior by considering variability of daily activities based on their occurrence time (e.g., day, weekdays, weekends) and applying a context-aware hierarchical clustering algorithm. Moreover, *Holmes* learns temporal relationships between multiple activities with the help of both sequential pattern mining and itemset mining algorithms. New scenarios can be added based on resident and expert's feedback to increase the accuracy of the system.

10.4 Methods and Technical Solutions

10.4.1 Problem Definition and a Use Case

Outlier detection refers to finding unexpected and abnormal patterns in data. The challenge in detecting outliers comes from the difficulty in defining a normal behavior, which includes the issue of labeling data [18]. Therefore, unsupervised learning methods or a combination of methods such as frequent pattern mining and clustering can be applied to analyze, understand and detect outliers. Finding unexpected patterns in video session data is challenging due to the scarcity of the labeled data.

We investigate a dataset of video sessions, where each video session

Table 10.1: Example of video sessions sorted by *Session ID* and *Date-time*

Session ID	Video ID	Date-time	Event type
1	002	Oct-01-16 22:44	client_roll
1	002	Oct-01-16 22:45	created
1	002	Oct-01-16 22:46	connectivity_changed
1	002	Oct-01-16 22:47	bitrate_switched
1	002	Oct-01-16 22:48	started
1	057	Oct-01-16 22:55	program_changed
1	057	Oct-01-16 23:22	pause
1	057	Oct-01-16 23:48	stopped
2	105	Oct-03-16 17:26	client_roll
2	105	Oct-03-16 17:27	created
2	105	Oct-03-16 17:28	connectivity_changed
2	105	Oct-03-16 17:29	bitrate_switched
2	105	Oct-03-16 17:30	bitrate_switched
2	105	Oct-03-16 17:31	bitrate_switched
2	105	Oct-03-16 17:32	stopped

consists of session ID, video ID, date and time of an occurring video event together with its type. The aim is to use frequent sequential pattern mining on sequences of video events to find unexpected or abnormal patterns of video events. Table 10.1 shows two examples of video sessions. Every session starts with a viewer logging into his/her account (*client_roll*), instantiating the video player (*created*) and ending with *stopped*. We denote an itemset

Table 10.2: Event types and their corresponding IDs

Event ID	Event type	Event ID	Event type
1	bitrate_switched	8	paused
2	buffering_started	9	play
3	buffering_stopped	10	program_changed
4	client_roll	11	scrubbed
5	connectivity_changed	12	started
6	created	13	stopped
7	error_occurred		

i by $(i_1, i_2, ..., i_j..., i_n)$, where each i_j is an item. Table 10.2 shows all the available event types that can appear in a video session together with their unique *ID*. A sequence α is an ordered list of itemsets and defined as $\langle a_1, a_2, ..., a_j, ..., a_m \rangle$, where each a_j is an itemset. In our case each itemset, a_j, is a singleton. Table 10.3 shows how the information of Table 10.1 can be summarized as a sequence of events for each viewer. Using the sequential pattern mining, we would like to find frequent sequential patterns in our

data, group them into clusters based on their similarities, and then each infrequent sequential pattern can be analyzed and matched to these clusters to find normal and abnormal patterns.

Table 10.3: Example of video sessions with sequences of events

Session ID	Video ID	Date-time	Event seq
1	002,057	Oct-01-16 22:44	$\langle 4, 6, 5, 1, 12, 10, 8, 13 \rangle$
2	105	Oct-03-16 17:26	$\langle 4, 6, 5, 1, 1, 1, 13 \rangle$

Our use case relates to analyzing a sudden increase in the number of video streaming performance events during video sessions. Performance changes in video streams are often reflected by the re-buffering and quality adaptation events (*buffering_started*, *buffering_stopped*, and *bitrate_switched*). A sudden increase in occurrence of such events can be related to some kind of performance issues at the system level. Considering only the total number of re-buffering and bitrate adaptation events, however, may not be a true indicator of a sudden change in overall performance of the video sessions. It may happen that the number of initiated sessions surge during a certain time interval and that results in an increase in *buffering_started*, *buffering_stopped* and *bitrate_switched* events. This is because every session normally has some buffering and bitrate change events. However, what is more important for the OVSPs is to identify if such event types within sessions increase in number for many concurrent video sessions and for many users approximately at the same time.

10.4.2 Clustering Analysis

Cluster analysis is a process of partitioning a set of objects into groups of similar objects. That is, the objects within each cluster are similar to each other but dissimilar to objects in neighboring clusters [19].

In our experiment, we use two different clustering methods to partition the data, namely *k-means* [20] and *affinity propagation (AP)* [21]. The popular k-means algorithm begins by an initial set of randomly selected centroids. It then iteratively revises this set until the sum of squared errors are minimized. k-means requires the value of k, i.e., the number of clusters, as an input.

Affinity propagation, on the other hand simultaneously considers all data points as potential centroids and exchanges real-valued messages between data points until a good set of centroids and clusters appear. The exchanged messages represent either the suitability of one data point in comparison to others being the centroid (responsibility) or when one data point should choose a new centroid (availability). AP adapts the number of clusters based on the data. In comparison with k-means, the AP algorithm uses actual data points as the cluster's centroids.

10.4.3 Cluster Validation Measures

The cluster validation techniques can be regarded as important aids for interpreting partitioning solutions to find the one that best fits the underlying data. Cluster validation measures can be divided into two major categories: external and internal. External validation measures require the ground truth labels for providing an assessment of clustering quality. In case the ground truth labels are not known, internal validation measures can be used. Internal measures base their analysis on the same information used to create the model itself. In general, internal measures can be used to assess compactness, separation, connectedness, and stability of the clustering results [22]. A detailed overview of different clustering validation measures and their comparison can be found in [23, 24].

In this study we apply *Silhouette Index (SI)* [2] as an internal validation measure due to unavailability of the ground truth labels. SI can be applied to evaluate the tightness and separation of each cluster and it measures how well an object fits the available clustering. For each i, let $a(i)$ be the average dissimilarity of i to all other objects in the same cluster. Let us now consider $d(i, C)$ as an average dissimilarity of i to all objects of a cluster C. After computing $d(i, C)$ for all clusters, the one with the smallest average dissimilarity is denoted as $b(i)$. Such cluster also refer to *neighboring cluster* of i. The Silhouette Index score of i, $s(i)$, is obtained by combing $a(i)$ and $b(i)$ as follows:

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

The $s(i)$ has values in a range of $[-1, 1]$. A score close to 1 implies that the object is well clustered. When $s(i)$ is about zero, this indicates the

object is on the decision boundary between two neighboring clusters. The worst situation occurs when $s(i)$ is close to -1. This indicates that the object is misclassified and assigned to the wrong cluster.

The average $s(i)$ for all objects i belonging to the same cluster shows how tightly those objects are grouped. The average $s(i)$ for all objects i in the whole dataset judges the quality of the generated clustering solution.

10.4.4 Distance Measures

In order to calculate the similarity between the frequent patterns with different lengths, we study two different distance measures, namely *Fast Dynamic Time Warping (FastDTW)* algorithm [25] and *Levenshtein Distance (LD)* [26].

The FastDTW algorithm is able to detect an accurate optimal alignment between two time series and to find the corresponding regions between them. FastDTW reduces the resolution of the time series repeatedly with averaging adjacent pairs of points. Then it takes a minimum-distance warp path at a lower resolution and projects to a higher resolution. The projected warp path is refined and repeatedly projected onto incrementally higher resolutions until a full warp path is found.

The LD, also known as edit distance, is a string similarity metric that measures the minimum number of editing operations (insertion, deletion and substitution) required to change one string into the other.

As mentioned earlier, the video session data has a temporal order, which means that events can only appear in a special set-up. Therefore, FastDTW is chosen for comparison of the patterns. On the other hand, LD as an edit distance considers the elements' alignments of the patterns and the required changes to transform one into other. As an example consider these two patterns, $P_1 : \langle 1, 1, 1, 1, 1 \rangle$ and $P_2 : \langle 1, 1, 1 \rangle$. According to FastDTW, these two patterns are 100% similar since the measure assumes that P_2 is bent. However, LD would show that the similarity between the two patterns are 60% since the insertion of 2 extra 1's are needed to transform P_2 to P_1. From the point of view of time series analysis these two patterns are similar. However, from the video streaming performance point of view, repetition of re-buffering and quality adaptation events may represent performance issues, which in this scenario the result of the LD is more relevant. For this reason, in this study we evaluate the proposed approach with both distance

measures (FastDTW and LD).

10.5 Proposed Approach

We combine frequent sequential pattern mining with clustering and Silhouette Index based analysis to detect unexpected patterns in online video data. Our approach can be found similar to Rossi et al.'s proposed method that has been applied for smart grid data in the district heating domain [16]. Both approaches deal with sequences of event types. However, instead of *itemset mining* we use *sequential pattern mining* due to the fact that the temporal order of occurrence of the video events is important.

To analyze video sessions for finding unexpected patterns at the system level the following steps are carried out:

1. *Data segmentation.* The video sessions are first divided into equal-sized segments based on the time period they are instantiated in order to identify sequential patterns. Data segmentation can be performed hourly, daily, and even weekly with different set-ups. For example, daily video sessions can be divided into four 6-hour period segments. Due to availability of daily patterns in the data, similar segments of similar days can be compared. We have conducted some initial experiments of our approach with bigger segment sizes, such as 2-days and weekly. However, additional evaluation and validation of these scenarios are needed to be able to make an informed conclusion about their significance for the approach performance. Therefore, in this paper, we have only considered a daily segment. One segment includes all video sessions that are initiated at the same time period.

2. *Frequent sequential patterns finding.* The PrefixSpan algorithm [13] is used to find frequent sequential patterns in each segment. The extracted patterns are stored in a list corresponding to each segment. These patterns can lead us to find *collective outliers*[1]. Note that we only use sequences of video events as inputs for the algorithm. Moreover, each video session has only one event sequence, such as $\langle 4, 6, 5, 1, 12, 10, 8, 13\rangle$ (see *Session ID 1* in Table 10.3). Those sequential patterns that satisfy the user-specified support will be stored as frequent patterns. In this study, the user-specified support threshold is set to be 0.15, which means any pattern that appears

[1] Collective outlier is a collection (sequence) of related data points that deviate significantly from the entire data set. Note that the individual data points in the sequence may or may not be outliers by themselves[18, 19].

more than $(0.15 * size\_of\_the\_segment)$ times will be considered. The support threshold is tested with different sizes ranging between 0.1 to 0.2. By choosing values close to 0.1 many patterns are extracted which affects the execution time dramatically. On the other hand, choosing values close to 0.2 ends up extracting very few patterns. However, by setting the support threshold to 0.15 we both decrease the execution time and gain a reasonable amount of patterns. Additionally, in order to decrease the computational time of the proposed approach, patterns with lengths less than 3 are omitted.
3. ***Frequent sequential patterns mapping.*** The list of extracted frequent sequential patterns is created for each segment in Step 2. Now, for each segment, the following steps will be carried out:

1. Select a pattern, one at a time, from the list of frequent sequential patterns and mark those video sessions that contain the pattern. Note that a video session can be matched with different patterns.

2. Store the date, the pattern(s) and its related length and frequency in the *selected_patterns* list (if the pattern does not match any video sessions its frequency will be set to 0). We can also add additional information here such as whether the pattern happened during working days, weekends or irregular days (e.g., public holidays), and what day-of-week, that can be helpful for finding a *contextual outlier*[2].

3. If not all patterns are selected, go back to 1 and select the next pattern.

After Step 3 the *selected_patterns* list contains the following details: 1) date, 2) pattern, 3) length of the pattern, 4) frequency of the pattern in the segment, 5) date-time information (e.g., day-of-week (Mon = 0, Tue = 2, ..., Sun = 6), and type-of-day (irregular day = 2, workday = 1, and weekend = 0)). Therefore, the *selected_patterns* list can represent one element according to Table 10.3 as [*date:* Oct-01-2016, *sequence:* $\langle 4, 6, 5, 1 \rangle$, *length:* 4, *frequency in segment:* 1, *day-of-week:* 6, *type-of-day:* 0].

4. ***Most frequent and non-most frequent patterns finding.*** At this step, we look for those sequential patterns that occurred in more than one segment, i.e., the *Most Frequent Sequential Patterns (MFSPs)*. The

[2] Contextual or conditional outlier is a data point that deviates significantly with respect to a specific context or condition [18, 19].

initial assumption is that frequent patterns that appear in more than one segment can be considered as normal. *Non-most Frequent Sequential Patterns (NMFSPs)* on the other hand can be assumed as potentially unexpected at this stage.

5. *MFSPs clustering.* The *selected_patterns* list summarizes detailed information regarding all video sessions. Then a clustering algorithm (e.g., k-means) can be used to group MFSPs into clusters. Note that since every video event has an ID (see Table 10.2), a sequential pattern such as {*client_roll*, *created*,*connectivity_changed*, *bitrate_switched*} can be transformed to $\langle 4, 6, 5, 1 \rangle$.

6. *Analysis of NMFSPs and outlier detection.* The clustering model built in the previous step can be used to analyze the NMFSPs, i.e., by matching each NMFSP into the MFSPs clustering model we can evaluate how well it fits into the model. The goodness-of-fit of a NMFSP can be identified by applying some internal cluster validation measures such as Silhouette Index. That is, those NMFSPs with Silhouette scores, $s(i)$, less than the average $s(i)$ for the whole clustering solution can be defined as outliers. Note that $s(i)$ measures how well an object i, a NMFSP in our case, fits the available clustering and ranges from -1 to 1. The Silhouette score close to 1 implies that the pattern is well clustered. When s_i is about zero, this indicates the NMFSP is on the decision boundary between two neighboring clusters. An $s(i)$ close to -1 indicates that the NMFSP is misclassified and assigned to an erroneous cluster, i.e., such NMFSP can be identified as an outlier.

10.6 Empirical Evaluation

10.6.1 Data Collection

We used two months of data (October-November 2016) for initial evaluation of sequential pattern mining to find unexpected patterns in video sessions. The data is obtained from a large European telecommunication company and contains 202,312 unique video session IDs, 2,213,330 events, 13 event types and 47,938 videos. Table 10.4 summarizes detailed information about the data for each month.

Table 10.4: Summary of the data used in the experiment

	October 2016	November 2016
No. of video session IDs	114,407	87,905
No. of events	1,327,679	885,651
No. of video IDs	26,266	21,672
No. of Event types	13	13

Table 10.5: The results of the experiment

			Affinity Propagation		*k-means*	
			LD	FastDTW	LD	FastDTW
Oct 2016	*No. of MFSPs*	384				
	No. of NMFSPs	60				
	SI		0.149	0.170	**0.182**	**0.203**
	No. of clusters		32	33	22	22
	No. of detected outliers		33	31	**40**	**36**
	No. of days		2 (31)	2 (31)	2 (31)	2 (31)
	No. of matched video sessions / day		143 (4,359)	144 (4,359)	**144** (4,359)	**402** (4,359)
			372 (2,390)	**372** (2,390)	336 (2,390)	336 (2,390)
Nov 2016	*No. of MFSPs*	109				
	No. of NMFSPs	258				
	SI		0.175	0.192	**0.194**	**0.207**
	No. of clusters		14	14	12	12
	No. of detected outliers		120	144	**137**	**160**
	No. of days		1 (30)	1 (30)	1 (30)	1 (30)
	No. of matched video sessions / day		1,068 (3,705)	1,078 (3,705)	1,068 (3,705)	1,078 (3,705)

Note. Numbers inside the parentheses represent the total for both days and video sessions.

10.6.2 Experimental Design

The proposed approach is implemented in Python version 3.6. The Python implementation of PrefixSpan, LD and FastDTW algorithms are fetched from [27–29] respectively. The clustering algorithms are adopted from the scikit-learn module [30]. The implemented code and the experimental results are available at GitHub[3].

[3] https://github.com/shahrooz-abghari/Outlier-Detection-for-Video-Session-Data-Using-Sequential-Pattern-Mining

In this study, we have investigated the usage of two different distance measures namely, LD and FastDTW together with two clustering methods and sequential pattern mining for detecting outliers. The motivation behind this is due to the fact that these distance measures are able to capture different similarity characteristics between the two compared patterns (see the discussion in Section 4.4 for more details).

We use SI to determine the optimal number of clusters on the set of MFSPs. Namely, we have run k-means algorithm with a different number of clusters. Then we have used the SI as a validity index to identify the best partitioning scheme. Figure 10.1 shows the average Silhouette scores for all k in the range between 2 and 35 using the LD (red color line) and FastDTW (blue color line) measures for data belongs to October 2016. The selected range is based on the number of clusters chosen by AP. We search for a local maximum of each plot that has a sudden change in order to identify the optimal k. The black box in Figure 10.1 shows the selected optimal $k = 22$, which is the same for both measures in October 2016. The optimal k for data belongs to November 2016, is 12. In addition, SI is also applied to analyze the NMFSPs.

10.7 Results and Analysis

The proposed approach is evaluated separately on data collected from October and November 2016. Two different clustering algorithms, AP and k-means together with LD and FastDTW are used for partitioning the MFSPs. The Silhouette Index is used to analyze NMFSPs on both clustering models. The results are presented in Tables 10.5, 10.6 and 10.7. As shown in Table 10.5, the number of extracted patterns, both MFSPs and NMFSPs, for October compared to November varies considerably. In October, the daily segments contain higher number of MFSPs, i.e., 384 compared to November which is 109. On the other hand, the number of NMFSPs for November is 258, which is approximately 4 times larger than the extracted patterns for October. This is mainly because the total number of video sessions and the frequency of event types in October is larger compared to November.

As presented in Table 10.5, during October and November both clustering algorithms detect outliers. In October we identify outliers in two days and in November only in one day. The combination of k-means algorithm with either of LD or FastDTW detected slightly more unexpected patterns compared to

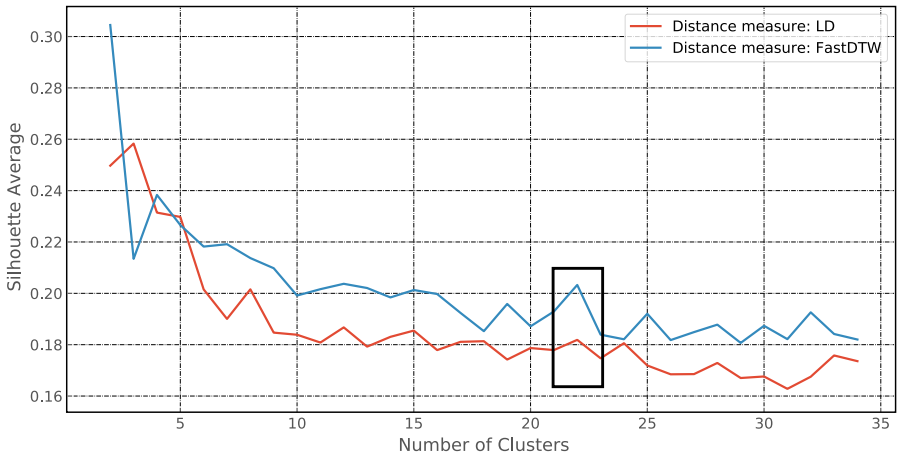
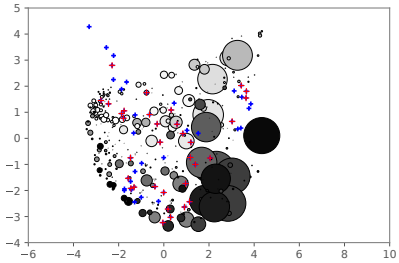


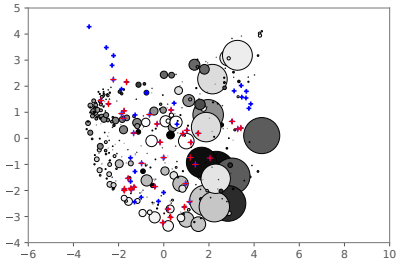
Figure 10.1: Identifying the optimal number of clusters for k-means using Silhouette Index for data belongs to October 2016
. The black box shows the selected optimal k for the studied distance measures.

AP. The number of video sessions that matched with the detected outliers by the clustering algorithms are quite similar in both months except for the combination of k-means and FastDTW, which hits 402 video sessions in October. This perhaps relates to how each algorithm selects centroids of the clusters and tries to minimize the sum of squared errors. Table 10.5 also presents the number of video sessions that match with detected outliers. Overall, k-means matched more video sessions with the identified outliers during the months of October and November.

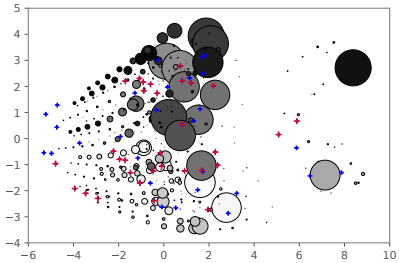
The results of the top five most frequent sequential patterns for both October and November 2016 are presented in Table 10.6. These patterns relate to the daily segment. These patterns are matched with the majority of the video sessions (101,996 out of 114,407 and 84,494 out of 87,905 matched video sessions for October and November 2016, respectively). Most of these patterns begin with created ($ID = 6$) and client_roll ($ID = 4$) followed by connectivity_changed ($ID = 5$), bitrate_switched ($ID = 1$) and started ($ID = 12$) events. These sequences of the video events are the most common ones. Moreover, three of these sequences contain paused ($ID = 8$) and stopped ($ID = 13$), which represent a complete video session that begins with a viewer's login and ends with *stopped*. The bold patterns in Table



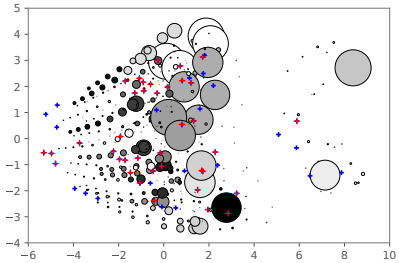
(a) AP and LD, (Oct 2016)



(b) k-means and LD, (Oct 2016)



(c) AP and FastDTW, (Oct 2016)



(d) k-means and FastDTW, (Oct 2016)

Figure 10.2: The visualization of the data is preformed by applying *Principal Component Analysis (PCA)* to convert the multi-dimensional dissimilarity matrices into 2-dimensional arrays. Therefore, no labels for axes are given. Each sphere represents one MFSP. The size of a sphere shows the number of video sessions that have been matched with it. Spheres with the same color belong to one cluster. The NMFSPs are shown with the blue pluses (" + ") and those NMFSPs that are identified as outliers are the red pluses (" + ").

10.6 represent those patterns that occur in both months and in the case of MFSPs they cover a high proportion of the video sessions.

Table 10.7 shows the top 5 NMFSPs detected as outliers by clustering algorithms. There are two patterns detected with both AP and k-means in October. The first pattern contains bitrate_switched ($ID = 1$), buffering_started ($ID = 2$), followed by two bitrate_switched ($ID = 1$) events. The second pattern contains client_roll ($ID = 4$) followed by connectivity_changed ($ID = 5$), bitrate_switched ($ID = 1$), started ($ID = 12$) and program_changed ($ID = 10$) events. There are three bold patterns, only one detected by AP and two by k-means. The pattern

Table 10.6: Top 5 most frequent sequential patterns (MFSPs) relate to the daily segment

	Pattern	Oct 2016	Pattern	Nov 2016
MFSP	$\langle \mathbf{6}, \mathbf{4}, \mathbf{5}, \mathbf{1}, \mathbf{12}, \mathbf{8}, \mathbf{13} \rangle$	74,362	$\langle \mathbf{6}, \mathbf{4}, \mathbf{5}, \mathbf{1}, \mathbf{12}, \mathbf{8}, \mathbf{13} \rangle$	59,340
	$\langle \mathbf{6}, \mathbf{4}, \mathbf{5}, \mathbf{1}, \mathbf{12} \rangle$	12,061	$\langle \mathbf{6}, \mathbf{4}, \mathbf{5}, \mathbf{1}, \mathbf{12} \rangle$	19,104
	$\langle 6, 4, 5, 1, 12, 1 \rangle$	6,491	$\langle \mathbf{6}, \mathbf{4}, \mathbf{5}, \mathbf{1}, \mathbf{12}, \mathbf{8} \rangle$	4,449
	$\langle \mathbf{6}, \mathbf{4}, \mathbf{5}, \mathbf{1}, \mathbf{12}, \mathbf{8} \rangle$	5,057	$\langle 6, 5, 1, 12, 8, 13 \rangle$	1,098
	$\langle 6, 4, 5, 1, 12, 1, 8, 13 \rangle$	4,025	$\langle 6, 5, 1, 12 \rangle$	503
Total matched patterns		101,996		84,494

Note. Highlighted patterns represent those that occur in both Oct and Nov 2016.

$\langle 1, 12, 10, 8 \rangle$, which is detected by k-means is a sub-pattern of $\langle 1, 12, 10, 8, 13 \rangle$ detected by AP and they are quite similar. However, this pattern $\langle 1, 2, 1, 1 \rangle$ detected by k-means using FastDTW is interesting mostly because it has repetition of bitrate_switched $(ID = 1)$ and matched with 256 video sessions. In general, every video session contains a number of re-buffering and bitrate switched. However, any increase in the quantity of such event types for many viewers can be related to performance issues. This follows the definition of a collective outlier, i.e., an unexpected collection or sequence of related event types (data points) occurring together. Nevertheless, more investigation needs to be performed to find the reason of these issues. For November, both the clustering algorithms detect the same number of patterns.

The results of applying the proposed approach on data belonging to October are visualized in Figure 10.2. Both AP and k-means detect outliers in two weekdays. In all plots each sphere represents one MFSP. The size of a sphere shows the number of video sessions that have been matched with it. The spheres with the same color belongs to one cluster. The NMFSPs are shown with blue " + " and the detected outliers displayed with red " + ". Principal Component Analysis (PCA) is used to transform the multi-dimensional dissimilarity matrices created by distance measures into 2-dimensional arrays. Plot (a) shows the results of AP using LD measure. As it is shown in Table 5, AP partitioned the MFSPs into 32 clusters. The results of k-means using LD measure is shown in (b). The size of k is set to be 22 and 40 NMFSPs out of 60 are marked as outliers. The plots (c) and (d) present the results of AP (no. of clusters = 33) and k-means (no.

Table 10.7: Top 5 non-most frequent sequential patterns (NMFSPs) detected as outliers for each month

		Affinity Propagation		*k-means*	
		LD	FastDTW	LD	FastDTW
Oct 2016	$\langle \mathbf{1}, \mathbf{2}, \mathbf{1}, \mathbf{1} \rangle$	-	-	-	256
	$\langle 1, 2, 1, 1, 8, 13 \rangle$	101	101	101	101
	$\langle \mathbf{1}, \mathbf{12}, \mathbf{10}, \mathbf{8} \rangle$	-	-	148	148
	$\langle \mathbf{1}, \mathbf{12}, \mathbf{10}, \mathbf{8}, \mathbf{13} \rangle$	136	136	-	-
	$\langle 4, 5, 1, 12, 10 \rangle$	152	152	140	140
	Total matched patterns	389	389	389	645
Nov 2016	$\langle 1, 1, 1, 1, 1 \rangle$	513	513	513	513
	$\langle 1, 2, 1 \rangle$	92	92	92	92
	$\langle 1, 1, 1, 2 \rangle$	67	67	67	67
	$\langle 3, 8, 13 \rangle$	66	66	66	66
	$\langle 1, 1, 1, 1, 2 \rangle$	53	53	53	53
	Total matched patterns	791	791	791	791

Note. '-' means unavailable.

of clusters = 22) using FastDTW measure, respectively. Using FastDTW measure, AP and k-means identified 31 and 36 outliers, respectively.

Using LD measure, it appears the partitioned MFSPs especially the bigger spheres, are condensed in clumps, surrounding the NMFSPs. However, with the FastDTW algorithm, MFSPs seem to be stacked vertically and the NMFSPs extend across the 2D space in a horizontal trend. These differences reflect how each distance measure calculates the dissimilarity between two patterns. As we mentioned earlier in Section 4.4, LD is more sensitive when part of one pattern is a sub-pattern of the other with different length. Nevertheless, using different distance measures together with 3D visualization techniques can provide a better understanding of the underlying organization of the data for OVSPs.

10.8 Discussion

Outlier detection approaches can assist the online video service providers to monitor and improve the quality of their services. In general, finding outliers in online video data without having a clear definition of normal behavior is a challenging task. Hybrid approaches combining sequential pattern mining and clustering analysis, as it has been demonstrated in this study, can be

useful in detecting unexpected sequences of events. In this study, video session data contains 13 unique event types (see Table 10.2 for more details). These event types are quite general and most of them can appear in both video sessions with *good* and *bad* quality. This makes it hard to draw any conclusions about the detected outliers without experts' validation. However, looking at the ratio of the quality related events can assist us to judge the quality of the video sessions. For example, by being able to identify a sudden increase of re-buffering and bitrate switch events, one may prevent users from having an unsatisfactory experience. The proposed approach has been evaluated with two months of data supplied by a large European telecommunication company. A number of outliers have been identified and matched with the studied use case, which analyzes a sudden increase of the video streaming performance events.

Perhaps it is worth to further study whether the different length of segments (see the discussion in Section 5) can affect the performance of the proposed outlier detection approach. In addition, it will be interesting to take into account the time interval between occurrences of the events in the evaluation set-up.

Furthermore, it is worthwhile to mention that not every pattern created by the sequential pattern mining algorithms can be useful. Although, sequential pattern mining searches for ordered sequences of events that are frequently happening together, some of the sequences might not be matched with any video sessions. The reason is that some of the events of the sequence are not available and pattern matching will not work. The importance of matching the extracted patterns with video sessions is due to the fact that we are trying to identify both the unexpected patterns and those sessions that are affected. Therefore, to ensure that no information will be lost, we plan to further study the combination of sequential pattern mining with frequent itemset mining.

10.9 Conclusion and Future Work

In this study, we have presented a hybrid approach for online video streaming by combining sequential pattern mining and clustering analysis to detect outliers at the system level. In addition, the usage of two different distance measures have been evaluated. In comparison to other studies that often apply statistical analysis to find outliers, we have looked for unexpected

patterns that can have impact on the video streaming performance.

By applying this approach, online video service providers can easily monitor suspicious video sessions and capture a better understanding about the viewers' experience.

For future work, we aim to pursue further evaluation and validation of our approach on a variety of datasets by applying alternative clustering analysis techniques, e.g., graph-based clustering approaches and different validation measures. Our future plans also involve integrating additional information into the analysis of non-most frequent sequential patterns supplied by the domain experts.

References

[1] S. S. Krishnan and R. K. Sitaraman. "Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs". In: *IEEE/ACM Transactions on Networking* 21.6 (2013), pp. 2001–2014.

[2] P. J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *J. of Computational and Applied Mathematics* 20 (1987), pp. 53–65.

[3] R. Agrawal, T. Imieliński, and A. Swami. "Mining association rules between sets of items in large databases". In: *Acm sigmod record*. Vol. 22. 2. 1993, pp. 207–216.

[4] J. Han, H. Cheng, D. Xin, and X. Yan. "Frequent pattern mining: Current status and future directions". In: *Data Mining and Knowledge Discovery* 15.1 (2007), pp. 55–86.

[5] R. Agrawal and R. Srikant. "Mining sequential patterns". In: *Proc. of the 11th Int'l Conf. on Data Engineering*. IEEE. 1995, pp. 3–14.

[6] C. Borgelt. "Frequent item set mining". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2.6 (2012), pp. 437–456.

[7] W. Shen, J. Wang, and J. Han. "Sequential pattern mining". In: *Frequent Pattern Mining*. Springer, 2014, pp. 261–282.

[8] R. Srikant and R. Agrawal. “Mining sequential patterns: Generalizations and performance improvements”. In: *Advances in Database Technology–EDBT’96* (1996), pp. 1–17.

[9] M. J. Zaki. “SPADE: An efficient algorithm for mining frequent sequences”. In: *Machine Learning* 42.1 (2001), pp. 31–60.

[10] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. “Sequential pattern mining using a bitmap representation”. In: *Proc. of the Eighth ACM SIGKDD Int’l Conf. on Knowledge Discovery and Data Mining*. 2002, pp. 429–435.

[11] J. Han, J. Pei, and Y. Yin. “Mining frequent patterns without candidate generation”. In: *ACM sigmod record*. Vol. 29. 2. 2000, pp. 1–12.

[12] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.-C. Hsu. “FreeSpan: Frequent pattern-projected sequential pattern mining”. In: *Proc. of the Sixth ACM SIGKDD Int’l Conf. on Knowledge Discovery and Data Mining*. 2000, pp. 355–359.

[13] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. “Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth”. In: *Proc. of the 17th Int’l Conf. on Data Engineering*. 2001, pp. 215–224.

[14] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. “Mining sequential patterns by pattern-growth: The PrefixSpan approach”. In: *IEEE Transactions on Knowledge and Data Engineering* 16.11 (2004), pp. 1424–1440.

[15] D. Barbará, Y. Li, J. Couto, J.-L. Lin, and S. Jajodia. “Bootstrapping a data mining intrusion detection system”. In: *Proc. of the 2003 ACM Symp. on Applied computing*. 2003, pp. 421–425.

[16] B. Rossi, S. Chren, B. Buhnova, and T. Pitner. “Anomaly detection in Smart Grid data: An experience report”. In: *Int’l Conf. on Systems, Man, and Cybernetics*. IEEE. 2016, pp. 002313–002318.

[17] E. Hoque, R. F. Dickerson, S. M. Preum, M. Hanson, A. Barth, and J. A. Stankovic. “Holmes: A comprehensive anomaly detection system for daily in-home activities”. In: *Int’l Conf. on Distributed Computing in Sensor Systems*. IEEE. 2015, pp. 40–51.

[18] V. Chandola, A. Banerjee, and V. Kumar. "Anomaly detection: A survey". In: *ACM Computing Surveys* 41.3 (2009), p. 15.

[19] J. Han, J. Pei, and M. Kamber. *Data mining: Concepts and techniques*. Elsevier, 2011.

[20] J. MacQueen et al. "Some methods for classification and analysis of multivariate observations". In: *Proc. of the Fifth Berkeley Symp. on Mathematical Statistics and Probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[21] B. J. Frey and D. Dueck. "Clustering by passing messages between data points". In: *Science* 315.5814 (2007), pp. 972–976.

[22] A. K. Jain and R. C. Dubes. "Algorithms for clustering data". In: Englewood Cliffs, NJ, 1988.

[23] L. Vendramin, R. J. Campello, and E. R. Hruschka. "On the comparison of relative clustering validity criteria". In: *Proc. of the 2009 SIAM Int'l Conf. on Data Mining*. SIAM. 2009, pp. 733–744.

[24] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. "On clustering validation techniques". In: *J. of Intelligent Information Systems* 17.2-3 (2001), pp. 107–145.

[25] S. Salvador and P. Chan. "Toward accurate dynamic time warping in linear time and space". In: *Intelligent Data Analysis* 11.5 (2007), pp. 561–580.

[26] V. I. Levenshtein. "Binary codes capable of correcting deletions, insertions, and reversals". In: *Soviet physics doklady*. Vol. 10. 8. 1966, pp. 707–710.

[27] C. Gao. "PrefixSpan algorithm source code". In: (2015). URL: `https://github.com/chuanconggao/PrefixSpan-py`.

[28] B. Jain. "Edit distance algorithm source code". In: (-). URL: `https://www.geeksforgeeks.org/dynamic-programming-set-5-edit-distance`.

[29] K. Tanida. "FastDTW algorithm source code". In: (2017). URL: `https://github.com/slaypni/fastdtw`.

[30] 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". In: *J. of Machine Learning Research* 12 (2011), pp. 2825–2830.

11

A Minimum Spanning Tree Clustering Approach for Outlier Detection in Event Sequences

Shahrooz Abghari, Veselka Boeva, Niklas Lavesson, Håkan Grahn, Selim Ickin, Jörgen Gustafsson
In: The 17th IEEE International Conference on Machine Learning and Applications: Special Session on Machine Learning Algorithms, Systems and Applications, December, 2018, Orlando, Florida, USA (Accepted for publication.).

Abstract

Outlier detection has been studied in many domains. Outliers arise due to different reasons such as mechanical issues, fraudulent behavior, and human error. In this paper, we propose an unsupervised approach for outlier detection in a sequence dataset. The proposed approach combines sequential pattern mining, cluster analysis, and a minimum spanning tree algorithm in order to identify clusters of outliers. Initially, the sequential pattern mining is used to extract frequent sequential patterns. Next, the extracted patterns are clustered into groups of similar patterns. Finally, the minimum spanning tree algorithm is used to find groups of outliers. The proposed approach has been evaluated on two different real datasets, i.e., smart meter data and video session data. The obtained results have shown that our approach can be applied to narrow down the space of events to a set of potential outliers and facilitate domain experts in further analysis and identification of system level issues.

11.1 Introduction

Outlier detection has been studied and used to detect anomalous behavior in different domains. Outliers refer to data points that are significantly

different from the rest of populations. They can happen due to mechanical issues, fraudulent behavior, human error and if they are not identified may lead to uncontrollable situations. Outlier detection refers to the problem of finding unusual patterns in data or unknown behaviors in a system [1, 2].

In this paper, we deal with outlier detection in sequence datasets. A sequence dataset is a collection of sequences of events or elements listed, often with concrete notion of time [3]. Due to importance of the sequential ordering of the events, sequential pattern mining for finding interesting subsequences in sequence datasets was introduced in 1995 [4]. Sequential pattern mining has a broad application in different fields such as bio-informatics [5, 6], marketing [7], network security [8], telecommunication [9, 10], and text mining [11, 12]. Data in these domains is highly dimensional and sparse which makes the identification of outliers more complex [13].

In this study, an outlier is defined as a small cluster (with respect to the number of matched sequences in a sequence dataset) which is significantly different from most of the frequent sequential patterns.

We propose an unsupervised 3-step approach that applies sequential pattern mining, cluster analysis, and a minimum spanning tree (MST) algorithm on a sequence dataset. In the first step, sequential pattern mining is used to extract frequent sequential patterns from data. In the second step, a clustering algorithm is applied on the extracted patterns in order to group the similar patterns together. Partitioning the patterns makes it possible in the third step to identify groups of patterns as outliers rather than detecting outliers individually. Consequently, this can lead to time complexity reduction in the proposed approach. In the third step, similar to [14], a minimum spanning tree is built on the clustering solutions in order to find clusters of outliers. By removing the longest edge(s) of the MST, the tree will be transformed to a forest. The small sub-tree(s) with few number of clusters (nodes) and/or with smaller sized clusters can be identified as outliers. The initial assumption is: the sub-trees with fewer nodes and smaller size contain patterns that happen rarely. Therefore, the clusters in these sub-trees are small, far and different from the clusters in the bigger sub-trees. The process of removing the longest edge(s) of the MST can also be performed by considering a user-specified threshold. The detected clusters of outliers can supply domain experts with a better understanding of the system behavior and facilitate them in the further analysis by mapping the

detected patterns to the corresponding sequences. The proposed approach has been evaluated on smart meter data and video session data. The results of the evaluation on video session data has been discussed with the domain experts.

11.2 Background and Related Work

Outlier detection techniques have been studied and successfully applied in different domains. There exists a considerable number of studies that provide a comprehensive, and structured overview of the state-of-the-art methods and applications for outlier detection [1, 2, 15, 16]. This attention shows the importance of outlier detection techniques in different domains and the fact that they are domain-specific.

Outlier detection techniques can be classified into three groups based on the availability of the labeled data [1, 2]: **1)** In the absence of prior knowledge of the data, unsupervised learning methods are used to determine outliers. The initial assumption is that normal data represents a significant portion of the data and is distinguishable from faults or error; **2)** In the presence of labeled data, both normality and abnormality are modeled. This approach refers to supervised learning; **3)** Define what is normal and only model normality. This approach is known as semi-supervised learning since the algorithm is trained by labeled normal data, however, it is able to detect outliers or abnormalities. Semi-supervised outlier detections are more widely used compared to supervised techniques due to an imbalanced number of normal and abnormal labeled data.

The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis and grouping similar data into clusters. There are several clustering algorithms capable of detecting noise and eliminating it from the clustering solution such as DBSCAN [17], CRUE [18], ROCK [19], and SNN [20]. Even though such techniques can be used to detect outliers, the main aim of the clustering algorithm is to perform the partitioning task rather than identifying outliers.

This led to proposing clustering-based techniques that are capable of detecting: **1)** single-point outliers such as the application of Self Organizing Maps for clustering normal samples and identifying anomalous samples[21], and Expectation Maximization [22] for identifying the performance problems

in distributed systems or **2)** groups of outliers such as the intrusion detection proposed by [23].

The application of MST has been studied by researchers in different fields including cluster analysis and outlier detection [14, 24–27]. A two-phase clustering algorithm is introduced for detecting outliers by [14]. In the first phase, a modified version of the k-means algorithm is used for partitioning the data. The modified k-means creates $k+i$ clusters, i.e., if a new data point is far enough from all clusters (k, number of clusters defined by the user), it will be considered as a new cluster (the $(k+i)^{th}$ cluster where, $i > 0$). In the second phase, an MST is built where, the tree's nodes represent the center of each cluster and edges show the distance between nodes. In order to detect outliers, the longest edges of the tree are removed. The sub-trees with a few number of clusters and/or smaller clusters are selected as outliers.

Wang et al. [24] developed an outlier detection by modifying k-means for constructing a spanning tree similar to an MST. The longest edges of the tree are removed to form the clusters. The small clusters are regarded as potential ouliters and ranked by calculating a density-based outlying factor.

A spatio-temporal outlier detection for early detection of critical events such as flood through monitoring a set of meteorological stations is introduced in [27]. Using geographical information of the data, a Delaunay triangulation network of the stations is created. The following step limits the connection of nodes to their closest neighbors while preventing far nodes from being linked directly. In the next step, an MST is constructed out of the created graph. In the final step, the outliers are detected by applying two statistical methods to detect exactly one or multiple outliers.

In this study, we propose a 3-step outlier detection approach that is specifically developed for sequence datasets. In the first step, frequent sequential patterns are extracted. In the second step, the selected patterns are clustered using the affinity propagation (AP) algorithm. In the third step, a minimum spanning tree similar to the study of Jiang et al. [14] is used to identify small groups of clusters as outliers.

11.3 Methods and Techniques

11.3.1 Sequential Pattern Mining

Sequential pattern mining is the process of finding frequently occurring patterns in a sequence dataset. The records of the sequence dataset contain sequences of events that often have chronological order. As examples of sequence data we can refer to customer shopping sequences, biological sequences, and video session events sequences.

Let $\mathcal{I} = \{i_1, i_2, ..., i_m\}$ be a set of all items. A sequence α defined as $\langle a_1, a_2, ..., a_j, ..., a_m \rangle$, where a_j is an itemset. Each itemset a_j is a subset of $\mathcal{I}$ that its items happened at the same time. In this study, each itemset (a_j) is a singleton. A sequence $\alpha = \langle a_1, a_2, ..., a_m \rangle$ is a subsequence of $\beta = \langle b_1, b_2, ..., b_n \rangle$ if and only if there exist integers $1 \leq k_1 < k_2 < ... < k_m \leq n$ and $a_1 \subseteq b_{k_1}, a_2 \subseteq b_{k_2}, ..., a_m \subseteq b_{k_m}$ [4]. Given a sequence dataset $\mathcal{T} = \{s_1, s_2, ..., s_j, ..., s_n\}$, where s_j is a sequence of itemsets, the support for α is the number of sequences that contain α as a subsequence. A sequence α is called a frequent sequential pattern if its support is equal or greater than user-specified support threshold.

To extract frequent sequential patterns the PrefixSpan algorithm [28] is used. PrefixSpan applies a *prefix-projection* method to find sequential patterns. Given a sequence dataset $\mathcal{T}$ and a user-specified threshold, the dataset is first scanned in order to identify all frequent items in sequences. All of these frequent items are considered as length-1 sequential pattern. After that, the search space is divided into a number of subsets based on the extracted prefixes. At last, for each subset a corresponding *projected dataset* is created and mined recursively.

Pei et al. [29] show in their study that PrefixSpan has the best overall performance compared to other sequential pattern mining algorithms such as GSP [30] and SPADE [31]. Therefore, the PrefixSpan algorithm is used for extracting sequential patterns in this study.

11.3.2 Similarity Measure

In order to calculate the similarity between the frequent sequential patterns with different lengths, we use *Levenshtein distance (LD)* metric [32]. The LD, also known as edit distance, is a string similarity metric that measures the minimum number of editing operations (insertion, deletion and substitution)

required to change one string into the other. We used normalized LD, i.e., the result of each comparison is normalized by dividing it with the length of the longest pattern. The score ranges between 0 and 1. Score 0 implies 100% similarity between the two patterns and 1 represents no similarity. LD is a simple algorithm capable of measuring the similarity between patterns with different lengths. In this study since the extracted patterns can have different length we choose to use LD as a similarity measure.

11.3.3 Clustering Method

The extracted patterns are clustered by using affinity propagation algorithm [33]. AP works based on the concept of exchanging messages between data points until a good set of exemplars (the most representative of a cluster) and corresponding clustering solution appears. The exchanged messages at each step assists AP to choose the best samples as exemplars and which data points should choose those samples to be their exemplars. AP adapts the number of clusters based on the data. However, the number of clusters can be controlled by *preference* parameter. That is, a high value of the preference will cause AP to form many clusters, while a low value will lead to a small number of clusters. If the preference value is not provided the median similarity or the minimum similarity will be used.

Unlike most clustering algorithms such as k-means that requires the number of clusters as an input, AP is able to estimate the number of clusters based on the data provided. AP can create clusters of different shapes and sizes, and the exemplars (the selected data points) are the representative of the clusters [34]. These characteristics make AP a suitable clustering algorithm for the chosen datasets in this study.

11.3.4 The Proposed Approach

The proposed approach consists of a preprocessing step (*Data segmentation*) and 3 main steps: 1) *Sequential patterns mining*, 2) *Frequent sequential pattern clustering*, and 3) *MST building and outlier detection analysis* as follows:

0) ***Data segmentation.*** The data is first partitioned into equal-sized segments in order to identify sequential patterns. Due to availability of daily patterns in the data, similar segments of similar days can be

compared. In this paper, we have studied a daily segment.

1. ***Sequential patterns mining.*** The first step concerns the extraction of frequent sequential patterns and mapping them with records of a sequence dataset.

    a) ***Frequent sequential patterns finding.*** The PrefixSpan algorithm is used to find frequent sequential patterns from each segment. The extracted patterns can lead us to find sequences of events that based on their occurrence together assumed to be anomalous, also known as *collective* outliers [1]. Those sequential patterns that satisfy the user-specified support threshold will be stored as frequent patterns. Note that in sequential patterns the order of the events is important and by using the sequential pattern mining both the frequency and the order of events in the extracted patterns are taken into account.

    b) ***Frequent sequential patterns mapping.*** In the second step the extracted patterns are mapped with the source they come from and stored in a *selected_patterns* list. This relates to identifying those video sessions that contain the patterns or a device that the patterns are extracted from. The following step can lead us to find additional information about patterns such as pattern frequency and its occurrence time. The latter is also useful for finding a *contextual* or *conditional* outlier, i.e., a data point assumed to be anomalous only in a specific context [1].

2. ***Frequent sequential pattern clustering.*** Using the Levenshtein distance measure, as explained in sub-section *B*, the pairwise similarities between all patterns are calculated and the similarity matrix are constructed. The selected patterns are partitioned by applying affinity propagation on the created similarity matrix. Note, an advantage of using AP is that it can estimate the number of clusters from data.

3. ***MST building and outlier detecting.*** The third step includes two sub-steps concerning the construction of an MST, and the identification of sub-trees with the smallest size as outliers.

a) ***MST building.*** The exemplars of the clusters are used for building a complete weighted graph where vertices of the graph are exemplars of the clusters and edges are the distance between them (traversing weight). Using the MST algorithm, the aim is to determine a sub-set of edges that connect all the vertices together without any cycles that has the minimum total edge weight.

b) ***Outlier detection analysis.***

   i. The longest edge of the tree is removed. Note that there can be more than one edge to cut.

   ii. The constructed MST will be replaced by the created sub-trees, i.e., a single tree becomes a forest. Note that step (ii) can be repeated until the distance between nodes of the sub-trees become less or equal to a user-specified threshold. For example the threshold can be set to 0.5.

   iii. The sub-trees are ranked from smallest to largest based on the number of items they match within the sequence dataset. Following the definition of outliers that refers to patterns that happen rarely and sufficiently far away from other patterns [1], here the smallest sub-trees can be regarded as outliers.

11.4 Experimental Methodology

11.4.1 Datasets

The proposed approach is evaluated on two datasets namely, *smart meter* and *video session*. The first dataset contains smart meters recorded events data provided by Elektro Ljubljana, a power distribution company in Slovenia [35]. The provided data contains 10,739,273 records that logged different events generated by 117,944 smart meters between May and August 2017. Due to the high number of data, we only considered data from May 2017 and sampled 30 devices out of 85,776 without replacement. To decrease the chance of any bias two datasets are created through sampling and the experiment ran on each set separately. This led to selection of 28 unique devices for each dataset, i.e., in total 56 distinct devices and 2 identical devices are sampled. The event sequences generated by the sampled devices for datasets 1 and 2 contain 40 and 44 unique event types respectively. The datasets together contain 36 identical event types. Each of these event types

have an informative description and a unique ID that explain the status of a device at a specific time, e.g., '*Voltage OK L1*', '*Power down L2*' and '*Power up*'.

Table 11.1 summarizes detailed information about the smart meters dataset. Since daily activity of similar devices are monitored for a period of one month (May 2017), sequential patterns are extracted individually from devices in each daily segment. Therefore, the PrefixSpan user-specified support threshold (Step 1-a of the proposed approach) is set to be 1. Furthermore, to reduce the time complexity, only patterns with length between 2 to 7 are considered. On the other hand, if the interest is to know what kind of issues are mostly common between all devices, frequent patterns can be extracted from each segment and by considering all devices. That is, the PrefixSpan user-specified support threshold can set to be the minimum percentage that extracted patterns should appear in a segment.

Table 11.1: Summary of the smart meter data, May 2017

No. of devices		85,776
No. of recorded event logs		2,265,08
No of Event types		135
	Sampled dataset 1	Sampled dataset 2
No. of selected devices	30 (28)	30 (28)
No. of recorded event logs	28,369	54,555
No. of event types	40 (4)	44 (8)

Note. The distinct number of devices or events in each dataset are listed in the parentheses.

The second dataset contains one month (February 2018) of video session data. The data is obtained from a large European telecommunication company and contains 288,669 unique video session IDs, 4,983,090 events, 19 event types and 23,485 video programs. Examples of the event types in video session data are '*Playback.Aborted*', '*Playback.BitrateChanged*', and '*Playback.PlayerReady*'.

Table 11.2 summarizes detailed information regarding this dataset. Since viewers receive a unique video session ID each time they login into their accounts, we extract frequent patterns that are common between all sessions in each daily segment. For this purpose after having some preliminary tests and discussions with the experts the user-specified support threshold

is set to be 20%, i.e., the extracted patterns should at least appear in $(length\_of\_segment * 0.2)$ of the sessions. Moreover, we assume that the extracted patterns must contain at least 3 event types.

Table 11.2: Summary of the video session data, February 2018

No. of video session IDs	288,669
No. of events	4,983,090
No. of video IDs	23,485
No. of event types	19

In all datasets, each event type represents an item. In the smart meter data, event sequences represent an overall status of each device per day while in the video session data, event sequences contain the quality related events and actions that have been performed by the viewers during the sessions. Moreover, event sequences in both datasets contain itemsets with exactly one event in each, i.e., the itemsets in this study are singletons.

11.4.2 Implementation and Availability

The proposed approach is implemented in Python version 3.6. The Python implementations of PrefixSpan and LD measure are fetched from [36] and [37], respectively. The affinity propagation algorithm is adopted from the scikit-learn module [38]. For constructing, manipulating, and visualizing a minimum spanning tree the NetworkX package is used [39]. The NetworkX package uses Kruskal's algorithm [40] for constructing the MST. The implemented code and the experimental results are available at GitHub[1].

11.5 Results and Discussion

11.5.1 Smart meter dataset

We performed random sampling on the smart meter dataset and created two datasets with 30 devices in each. More details can be found in *Datasets* sub-section and in Table 11.1.

Applying the proposed approach on the first sampled dataset, leads to extracting 6,550 patterns. Using AP the collected patterns are partitioned

[1] https://github.com/shahrooz-abghari/MST-Clustering-Approach

into 253 clusters. Finally, by building the MST on top of the exemplars of the clustering solution and cutting the longest edge(s) three sub-trees are constructed. The two smallest sub-trees are identified as outliers. The patterns in these sub-trees are matched with 8 devices. Examples of detected patterns as outliers in daily operation of smart meters in May 2017 are as follows: {'Adjust time/date (old time/date)', 'Adjust time/date (new time/date)'}, {'Power down L3', 'Power down L1', 'Power down L2', 'Power restored L3', 'Power restored L1', 'Power restored L2'}, and {'Remote communication module OK', 'Communication board access error, PLC or GSM/GPRS module'}.

Table 11.3: The results of the experiment for smart meter data

Dataset 1	No. of extracted patterns	6,550
	No. of patterns detected as outlier	241
	No. of clusters	253
	No. of detected outliers	1 sub-tree with 1 node 1 sub-tree with 39 nodes
	No. of devices with issue	6
Dataset 2	No. of extracted patterns	6,676
	No. of patterns detected as outlier	215
	No. of clusters	211
	No. of detected outliers	2 sub-trees, 1 node each
	No. of devices with issue	10

Note. Numbers inside the parentheses represent unique devices.

For the second sampled dataset, in total 6,676 patterns are identified. The extracted patterns are partitioned in 211 clusters and removing the longest edge(s) of the MST led to identifying 2 sub-trees out of 3 as outliers. Fig. 11.1 (Top) shows the constructed MST, the created forest after cutting the longest edges (edges A and B) of the MST, and the detected sub-trees as outliers. The patterns in these two clusters are matched with 10 devices. Examples of the identified interesting sequential patterns are as follows: {'Wrong phase sequence', 'Wrong phase sequence'}, {'Under voltage L2', 'No under voltage L2 anymore'}, and {'Communication error with FLEX meter/Measuring system access error', 'Meter communication OK with FLEX meter/Measurement System OK', 'Communication error with FLEX meter/Measuring system access error'}. Table 11.3 summarizes the results of the experiment on the smart meter data.

The results of the experiment on the smart meter data showed that the

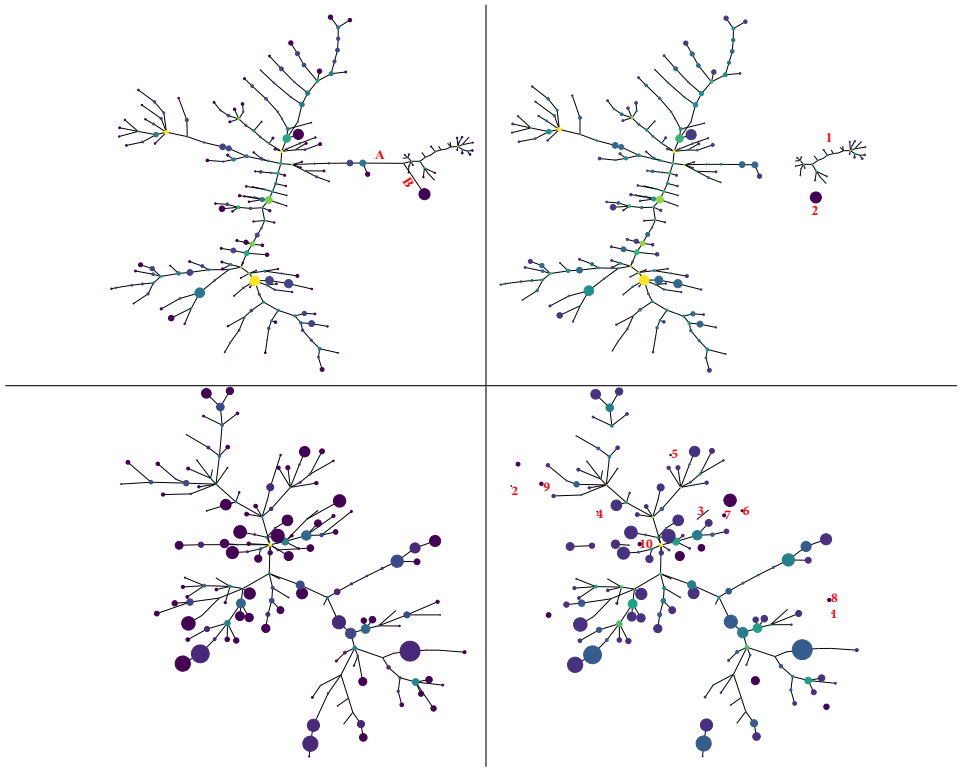
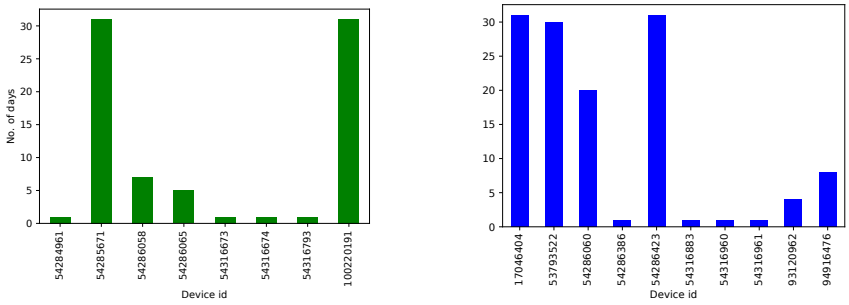


Figure 11.1: **(Top-left)** The constructed MST before removing the longest edges on smart meter sampled dataset 1. Edges A and B represent the longest edges of the tree. **(Top-right)** The transformation of the constructed MST into a forest with 3 sub-trees after the longest edges are removed. The sub-trees 1 and 2 are considered as outliers based on their size. **(Bottom-left)** The constructed MST before removing the longest edges on video session dataset. **(Bottom-right)** The transformation of the constructed MST into a forest with 22 sub-trees after the longest edges are removed. The sub-trees are ranked from smallest to largest based on their size. The top 10 smallest sub-trees are considered as outliers. ***Note.*** The size of a node represents the number of smart meters or video sessions that are matched with it. The color of a node shows the degree of the node and is used only for the visualization purposes. The distance between edges range between [0,1].

detected patterns as outliers are matched with 8 devices in the sampled dataset 1 and 10 devices in the sampled dataset 2. Fig. 11.2 shows the identified device ids with issues and the number of days they were faulty. Table 11.4 presents the top 5 sequential patterns that identified as outliers for each dataset. Perhaps only some of these patterns represent serious



(a) Dataset 1: 8 devices are identified with issues.

(b) Dataset 2: 10 devices are identified with issues.

Figure 11.2: Identified smart meter with issues for both sampled datasets 1 and 2, May 2017.

issues and some only relate to miss-configuration such as Adjust time/date. However, since such sequences of patterns occurred rarely in a normal daily activity of the monitored devices they have been detected as outliers. Nevertheless, further analysis and domain experts' opinion are needed to determine the severity of the issues raised by these patterns.

11.5.2 Video session dataset

We applied our proposed approach on one month (February 2018) video session data. In total 1,493 sequential patterns are mined. AP partitioned the patterns into 170 clusters and cutting the longest edges of the MST led to 22 sub-trees. We sort the sub-trees based on the number of video sessions they matched with, and choose the top 10 smallest sub-trees as outliers. Fig. 11.1 (Bottom) shows the constructed MST, the created forest after cutting the longest edges of the MST, and the top 10 smallest sub-trees identified as outliers. In total 10,121 video sessions are matched with patterns in these sub-trees. Examples of identified pattern as outlier are {'Playback.BufferingStarted', 'Playback.ScrubbedTo', 'Playback.Aborted'} and {'Playback.BitrateChanged', 'Playback.Resumed', 'Playback.Completed'}. Table 11.5 summarizes the results of the experiment on the video session data.

The results of the experiment on the video session dataset showed on average 3.5% of the sessions in each day contained outliers. Fig. 11.3

Table 11.4: Top 5 sequential patterns identified as outliers for smart meter sampled datasets, May 2017

	Pattern	Freq. of the pattern
Dataset 1	⟨165, 79, 165, 79, 79, 165, 165⟩	27
	⟨10, 11, 11, 10, 11, 10, 11⟩	15
	⟨20, 22⟩	3
	⟨21, 20⟩	3
	⟨397, 396, 393, 395, 392, 63, 346⟩	1
Dataset 2	⟨20, 20⟩	32
	⟨245, 245, 245, 245, 245, 245, 245⟩	31
	⟨63, 63, 63, 63, 63, 63, 63⟩	28
	⟨21, 21⟩	17
	⟨22, 184⟩	17

Note. Event names equivalent to each ID are as follows: **ID** = 10, *Adjust time/date (old time/date)*, **ID** = 11, *Adjust time/date (new time/date)*, **ID** = 20, *Over voltage on L1*, **ID** = 21, *Over voltage on L2*, **ID** = 22, *Over voltage L3*, **ID** = 63, *Wrong phase sequence*, **ID** = 79, *Communication board access error, PLC or GSM/GPRS module*, **ID** = 165, *Remote communication module OK*, **ID** = 184, *No over voltage L3 anymore*, **ID** = 245, *E Meter command error*, **ID** = 346, *Voltage L2 normal*, **ID** = 392, *Power down L1*, **ID** = 393, *Power down L2*, **ID** = 394, *Power down L3*, **ID** = 395, *Power restored L1*, **ID** = 396, *Power restored L2*, **ID** = 397, *Power restored L3*.

Table 11.5: The results of the experiment for video session data

No. of extracted patterns	1,493
No. of patterns detected as outliers	88
No. of clusters	170
No. of sub-trees	22
No. of detected outliers	1 sub-tree with 2 nodes 9 sub-trees, 1 node each
No. of video sessions with issue	10,121

shows the total number of video sessions against the number of detected sessions as outliers and their percentage. In days 1, 23, 24, 26, and 28 of February higher number of outliers are detected. In the video session data, it is hard to draw any conclusions regarding the detected patterns as outliers without experts' validation. Unlike smart meter data that each event shows explicitly the status of a device, the event types in video session data are more general and most events can appear in both sessions with

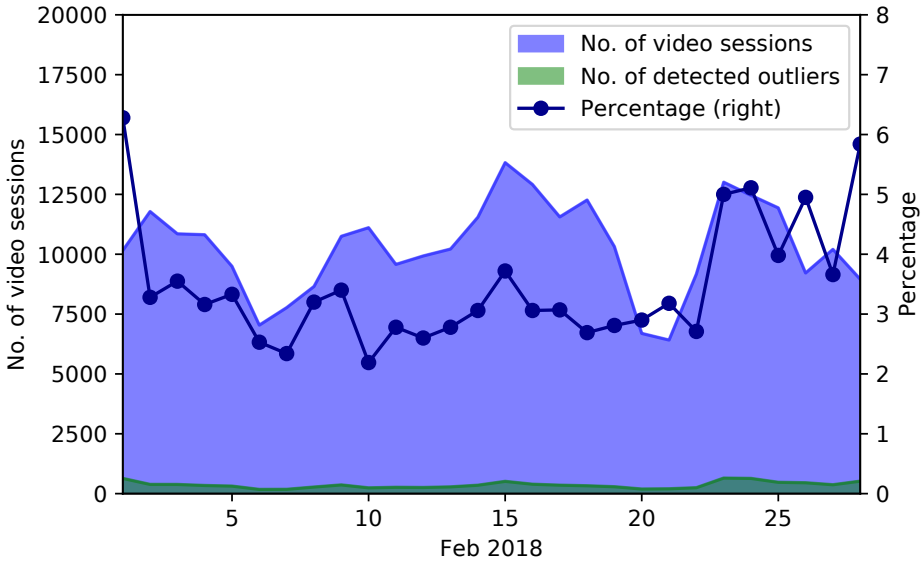


Figure 11.3: Visualization of the total number of video sessions vs. detected outliers for each day of February 2018 together with the percentage of the outliers.

good and *bad* quality. However, the ratio of quality related events such as *Playback.BitrateChanged*, $ID = 2$, and *Playback.BufferingStarted*, $ID = 4$ or *Playback.BufferingStopped*, $ID = 5$ in a viewer's session can assist us to judge the quality of the session. A sudden increase in any of these three events in a session and simultaneously for many viewers can represent an issue at the system level.

Table 11.6 shows the top 10 sequential patterns that identified as outliers and the number of video sessions that matched with them. Among these ten patterns seven of them (patterns 1-6 and 9) contain the quality related events that are mentioned earlier. On the other hand there are two patterns (7 and 9) that contain an event type *Playback.BufferingEnded*, $ID = 3$. This event often generates when the viewers scrub the video. Scrubbing is an action that helps a viewer to navigate through a video program to watch from a specific section. However, sometimes the viewers have to scrub the video due to frozen screen. Nevertheless, if the ratio of *Playback.BufferingEnded* increases inside a video session and at the same time for many sessions can relate to an issue at the system level. We have discussed the obtained results

Table 11.6: Top 10 sequential patterns identified as outliers for video session dataset, February 2018

No.	Pattern	Freq. of the pattern
1	$\langle \mathbf{2}, 16, 6 \rangle$	1613
2	$\langle 11, 7, \mathbf{2}, 18 \rangle$	838
3	$\langle 11, 17, \mathbf{4} \rangle$	653
4	$\langle 11, 7, \mathbf{2}, 16 \rangle$	477
5	$\langle 12, \mathbf{5}, 1 \rangle$	459
6	$\langle \mathbf{4}, 17, 1 \rangle$	445
7	$\langle 11, 3, 15 \rangle$	403
8	$\langle 18, 17, 1 \rangle$	401
9	$\langle 11, 3, \mathbf{4} \rangle$	298
10	$\langle 12, 14, 1 \rangle$	290

Note. The numbers in bold represent the quality related events. Event names equivalent to each ID are as follows: **ID** = 1, *Playback.Aborted*, **ID** = 2, *Playback.BitrateChanged*, **ID** = 3, *Playback.BufferingEnded*, **ID** = 4, *Playback.BufferingStarted*, **ID** = 5, *Playback.BufferingStopped*, **ID** = 6, *Playback.Completed*, **ID** = 7, *Playback.Created*, **ID** = 11, *Playback.HandshakeStarted*, **ID** = 12, *Playback.InitCompleted*, **ID** = 14, *Playback.PlayReady*, **ID** = 15, *Playback.PlayerReady*, **ID** = 16, *Playback.Resumed*, **ID** = 17, *Playback.ScrubbedTo*, **ID** = 18, *Playback.Started*.

with domain experts. In order to validate the results, the experts asked us to randomly select 18 video sessions (9 normal and 9 abnormal) from February 1, 2018. The labels of the video sessions were unknown for the expert. The validation shows only 3 video sessions can be considered as true anomalies and the other sessions are normal. This means $12/18 = 67\%$ of the video sessions are labeled correctly by the proposed approach. Further analysis of the experts' comments has revealed that assessing the quality of a video session is not an easy task and sometimes can be subjective. For example, the latter is supported by the experts' comments concerning the following 4 video sessions (V_1 to V_4) out of 6 that are mislabeled as outliers by the proposed approach:

V_1. "Short session (15,5 seec), no buffering, good bitrate, hard to tell, I tend to OK."

V_2. "No buffering, rather short, 10 sec, bitrate rather good. OK."

V_3. "Probably still OK considering the duration of the playback (1055 sec),

but some buffering (35 sec in total), with varying bitrate, I tend to OK."

V_4. "Some buffering, but below 0,5% of the play duration, played 612 sec, error event at the end of the session with no further explanation, played 10 minutes with good bitrate, I tend to OK."

The validation of the results has shown that the proposed approach is able to identify video sessions that are significantly different from the majority of the sessions due to occurrence of some specific event types. The identified anomalous sequential patterns can help the domain experts to understand the outlying properties of the detected outliers.

11.6 Conclusion

In this study, we have presented an outlier detection for sequence datasets. Our approach combines sequential pattern mining, clustering and minimum spanning tree to identify outliers. We have shown that the proposed approach can facilitate the domain experts in identification of outliers. Building the minimum spanning tree on top the clustering solution can lead to identifying clusters of outliers. This can reduce the time complexity of the proposed approach. Moreover, in this study we have looked into collective outliers, sequences of events that based on their occurrence together assumed to be anomalous, which may help to find the outlying properties of the detected outliers.

The proposed approach has been applied on two sequence datasets, smart meter data and video session data. Both datasets contain sequences of event types that either shows the operational status of a smart meter or the current action that takes place in a viewer's video session. The results of the experiments on the smart meters data are more comprehensible compared to the video session data. The main reason is the fact that the event types in smart meters are explicitly detailed, explaining the status of the devices. However, in video session data the event types are general which requires more investigation and experts' knowledge in order to detect video sessions with quality issues. The validation of the results on video session data by the domain experts showed that 67% of the labeled sessions by the proposed approach were correct.

For future work, we are going to further evaluate and validate the proposed approach by using different distance measures and clustering algorithms. These two parameters may have a strong correlation and it worths to further study how such correlation can affect the final results.

In this study, the MST is constructed from a complete weighted graph using the exemplars of the clusters. However, according to Cipolla et al.'s study a Delaunay triangulation network can be used to create a simplified graph. The Delaunay triangulation limits the number of connections of a node to its closest neighbors. This makes the constructed MST from this simplified graph a better representative compared to the complete graph. Therefore, we are also interested in testing whether the Delaunay triangulation network can be integrated into our approach.

References

[1] V. Chandola, A. Banerjee, and V. Kumar. "Anomaly detection: A survey". In: *ACM Computing Surveys* 41.3 (2009), p. 15.

[2] V. Hodge and J. Austin. "A survey of outlier detection methodologies". In: *Artificial Intelligence Review* 22.2 (2004), pp. 85–126.

[3] J. Han, H. Cheng, D. Xin, and X. Yan. "Frequent pattern mining: Current status and future directions". In: *Data Mining and Knowledge Discovery* 15.1 (2007), pp. 55–86.

[4] R. Agrawal and R. Srikant. "Mining sequential patterns". In: *Proc. of the 11th Int'l Conf. on Data Engineering*. IEEE. 1995, pp. 3–14.

[5] T. P. Exarchos, C. Papaloukas, C. Lampros, and D. I. Fotiadis. "Mining sequential patterns for protein fold recognition". In: *J. of Biomedical Informatics* 41.1 (2008), pp. 165–179.

[6] J. Guan, D. Liu, and D. A. Bell. "Discovering motifs in DNA sequences". In: *Fundamenta Informaticae* 59.2-3 (2004), pp. 119–134.

[7] Y.-L. Chen, M.-H. Kuo, S.-Y. Wu, and K. Tang. "Discovering recency, frequency, and monetary (RFM) sequential patterns from customers' purchasing data". In: *Electronic Commerce Research and Applications* 8.5 (2009), pp. 241–251.

[8] L.-C. Wuu, C.-H. Hung, and S.-F. Chen. “Building intrusion pattern miner for Snort network intrusion detection system”. In: *J. of Systems and Software* 80.10 (2007), pp. 1699–1715.

[9] F. Eichinger, D. D. Nauck, and F. Klawonn. “Sequence mining for customer behaviour predictions in telecommunications”. In: *ECML/PKDD 2006 Workshop on Practical Data Mining: Applications, Experiences and Challenges*. 2006, pp. 3–10.

[10] T. H. N. Vu, K. H. Ryu, and N. Park. “A method for predicting future location of mobile user for location-based services system”. In: *Computers & Industrial Engineering* 57.1 (2009), pp. 91–105.

[11] S. Jaillet, A. Laurent, and M. Teisseire. “Sequential patterns for text categorization”. In: *Intelligent Data Analysis* 10.3 (2006), pp. 199–214.

[12] P. C. Wong, W. Cowley, H. Foote, E. Jurrus, and J. Thomas. “Visualizing sequential patterns for text mining”. In: *Symp. on Information Visualization*. IEEE. 2000, pp. 105–111.

[13] C. C. Aggarwal and P. S. Yu. “Outlier detection for high dimensional data”. In: *ACM Sigmod Record*. Vol. 30. 2. ACM. 2001, pp. 37–46.

[14] M.-F. Jiang, S.-S. Tseng, and C.-M. Su. “Two-phase clustering process for outliers detection”. In: *Pattern Recognition Letters* 22.6 (2001), pp. 691–700.

[15] Y. Zhang, N. Meratnia, and P. Havinga. “Outlier detection techniques for wireless sensor networks: A survey”. In: *IEEE Communications Surveys & Tutorials* 12.2 (2010), pp. 159–170.

[16] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han. “Outlier detection for temporal data: A survey”. In: *IEEE Transactions on Knowledge and Data Engineering* 26.9 (2014), pp. 2250–2267.

[17] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. “A density-based algorithm for discovering clusters in large spatial databases with noise”. In: *KDD*. Vol. 96. 34. 1996, pp. 226–231.

[18] S. Guha, R. Rastogi, and K. Shim. “CURE: An efficient clustering algorithm for large databases”. In: *ACM Sigmod Record*. Vol. 27. 2. ACM. 1998, pp. 73–84.

[19] S. Guha, R. Rastogi, and K. Shim. "ROCK: A robust clustering algorithm for categorical attributes". In: *Proc. of the 15th Int'l Conf. on Data Engineering*. IEEE. 1999, pp. 512–521.

[20] L. Ertöz, M. Steinbach, and V. Kumar. "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data". In: *Proc. of the 2003 SIAM Int'l Conf. on Data Mining*. SIAM. 2003, pp. 47–58.

[21] F. A. González and D. Dasgupta. "Anomaly detection using real-valued negative selection". In: *Genetic Programming and Evolvable Machines* 4.4 (2003), pp. 383–403.

[22] X. Pan, J. Tan, S. Kavulya, R. Gandhi, and P. Narasimhan. "Ganesha: Blackbox diagnosis of mapreduce systems". In: *ACM SIGMETRICS Performance Evaluation Review* 37.3 (2010), pp. 8–13.

[23] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. "A geometric framework for unsupervised anomaly detection". In: *Applications of data mining in computer security*. Springer, 2002, pp. 77–101.

[24] X. Wang, X. L. Wang, and D. M. Wilkes. "A minimum spanning tree-inspired clustering-based outlier detection technique". In: *Ind. Conf. on Data Mining*. Springer. 2012, pp. 209–223.

[25] A. C. Müller, S. Nowozin, and C. H. Lampert. "Information theoretic clustering using minimum spanning trees". In: *Joint DAGM (German Association for Pattern Recognition) and OAGM Symp.* Springer. 2012, pp. 205–215.

[26] G.-W. Wang, C.-X. Zhang, and J. Zhuang. "Clustering with Prim's sequential representation of minimum spanning tree". In: *Applied Mathematics and Computation* 247 (2014), pp. 521–534.

[27] E. Cipolla, U. Maniscalco, R. Rizzo, D. Stabile, and F. Vella. "Analysis and visualization of meteorological emergencies". In: *Ambient Intelligence and Humanized Computing* 8.1 (2017), pp. 57–68.

[28] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth". In: *Proc. of the 17th Int'l Conf. on Data Engineering*. 2001, pp. 215–224.

[29] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. “Mining sequential patterns by pattern-growth: The PrefixSpan approach”. In: *IEEE Transactions on Knowledge and Data Engineering* 16.11 (2004), pp. 1424–1440.

[30] R. Srikant and R. Agrawal. “Mining sequential patterns: Generalizations and performance improvements”. In: *Advances in Database Technology–EDBT’96* (1996), pp. 1–17.

[31] M. J. Zaki. “SPADE: An efficient algorithm for mining frequent sequences”. In: *Machine Learning* 42.1 (2001), pp. 31–60.

[32] V. I. Levenshtein. “Binary codes capable of correcting deletions, insertions, and reversals”. In: *Soviet physics doklady*. Vol. 10. 8. 1966, pp. 707–710.

[33] B. J. Frey and D. Dueck. “Clustering by passing messages between data points”. In: *Science* 315.5814 (2007), pp. 972–976.

[34] U. Bodenhofer, A. Kothmeier, and S. Hochreiter. “APCluster: An R package for affinity propagation clustering”. In: *Bioinformatics* 27.17 (2011), pp. 2463–2464.

[35] Elektro Ljubljana. *Smart meters recorded events dataset*. 2018. URL: `https://data.edincubator.eu/organization/elektro-ljubljana-podjetje-zadistribucijo-elektricne-energije-d-d`.

[36] C. Gao. “PrefixSpan algorithm source code”. In: (2015). URL: `https://github.com/chuanconggao/PrefixSpan-py`.

[37] B. Jain. “Edit distance algorithm source code”. In: (-). URL: `https://www.geeksforgeeks.org/dynamic-programming-set-5-edit-distance`.

[38] 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”. In: *J. of Machine Learning Research* 12 (2011), pp. 2825–2830.

[39] A. Hagberg, P. Swart, and D. S Chult. *Exploring network structure, dynamics, and function using NetworkX*. Tech. rep. Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[40] J. B. Kruskal. “On the shortest spanning subtree of a graph and the traveling salesman problem”. In: *Proc. of the American Mathematical Society* 7.1 (1956), pp. 48–50.