h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

fbi
FACHBEREICH INFORMATIK

fbmn
FACHBEREICH MATHEMATIK
UND NATURWISSENSCHAFTEN

Hochschule Darmstadt

Fachbereiche Mathematik
und Naturwissenschaften
& Informatik

Aiding the Detection of Explosive Materials with Machine Learning: Extracting Relevant Features from Multivariate Sensor Data

Abschlussarbeit zur Erlangung des akademischen Grades
Master of Science (M. Sc.)
im Studiengang Data Science

vorgelegt von

Yannick Meuser

Referent:	Prof. Dr. Markus Döhring
Korreferentin:	Prof. Dr. Antje Jahn

Ausgabedatum:	05. Februar 2021
Abgabedatum:	20. Juli 2021

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Köln, 20. Juli 2021

Yannick Meuser

KURZFASSUNG. Eine zuverlässige und effektive Detektion von explosionsfähigen Substanzen ist eine der wichtigsten Aufgaben der zivilen Sicherheit. Oftmals als Gegenmaßnahme zu terroristischen Akteuren ist die Beseitigung eines potentiellen Explosionsstoffes ein essentielles Mittel der Gefahreneindämmung. Die Szenarien können vielfältig sein und reichen von der Bedrohung des öffentlichen Lebens an Flughäfen, Bahnhöfen oder Veranstaltungen, bis hin zur kritischen Infrastruktur höchster Sicherheit, wie z. B. Atomkraftwerke. Verfahren über die Bestimmung des Gefahrenpotentials eines unbekannten, verdächtigen Objekts sind vielfältig. Die Mindestanforderungen sind hierbei gleich hoch für alle: ein schneller Entscheidungsprozess in einer zeitkritischen Situation sowie Verlässlichkeit und ein hoher Abdeckungsgrad bei der Aufspürung verschiedenartiger Explosionsstoffe. Für dieses Anwendungsszenario wird in der Thesis die Weiterentwicklung eines bereits bestehenden Modells ausgearbeitet, das für die Detektion von explosionsfähigen Materialien dient. Der Anwendungsfall wird auf zwei weitere, neue Methodiken gelenkt: der Detektion hoch-energetischer Explosionsstoffe und der Identifizierung bestimmter Untergruppen, basierend auf der chemischen Struktur der Substanzen. Sensordaten aus einem speziellen Messverfahren für explosionsfähige und harmlose Stoffe werden für diesen Zweck verarbeitet. Der thematische Schwerpunkt wird auf die Extraktion von Merkmalen gelegt, für die Erstellung eines Prognosemodells. Ziel dieser Thesis ist es, den bestehenden Entscheidungsprozess der Detektion durch die Erweiterung neuer Anwendungsfälle sinnvoll zu ergänzen, um einen hohen Detailgrad über eine unbekannte, potentiell gefährliche Substanz geben zu können. Die Ergebnisse der Implementierung fallen für beide Anwendnugsfälle unterschiedlich aus. Während die Detektion von hoch-energetischen Substanzen gut möglich ist, ist bei der Identifizierung von Untergruppen nach chemischer Struktur noch Verbesserungspotential zu sehen.

Schlagworte Clustering ▪ Data Mining ▪ Energetische Materialien ▪ Feature Engineering ▪ Zeitreihenklassifikation

ABSTRACT. Reliable and effective detection of explosive substances is one of the most important tasks of civil security. Often as a countermeasure to terrorist actors, the elimination of a potential explosive is an essential means of hazard containment. Scenarios can be varied and range from threats to the public life at airports, train stations or events, to critical infrastructure of highest security, such as nuclear power plants. Procedures for determining the threat potential of an unknown, suspicious object are manifold. The minimum requirements are the same for all of them: a fast decision-making process in a time-critical situation, as well as reliability and a high degree of coverage in the detection of different types of explosives. For this application scenario, the thesis elaborates on the further development of an already existing model which is used for the detection of explosive materials. The use case is directed to two further, new methodologies: the detection of high-energetic explosive materials, and the identification of specific subgroups, based on the chemical structure of the substances. Sensor data from a special measurement method for explosive and harmless substances is processed for this purpose. The thematic focus will be on the extraction of features for the creation of a prediction model. The aim of this thesis is to extend the existing decision process of detection by implementing new use cases in a meaningful way, in order to achieve a high level of detail about an unknown, potentially dangerous substance. A difference in performance can be seen in the results for both use cases. While the detection of high-energetic substances is well possible, there is still potential for improvement in the identification of subgroups according to chemical structure.

Keywords Clustering ▪ Data Mining ▪ Energetic materials ▪ Feature Engineering ▪ Time series classification

Contents

List of Figures

List of Tables

Acronyms

BOSS	Bag of Symbolic Fourier Approximation Symbols
BoW	Bag of Words
CRISP-DM	CRoss-Industry Standard Process for Data Mining
DBA	DTW Barycenter Averaging
DFT	Discrete Fourier Transform
DTW	Dynamic Time Warping
KM	K-Means
LDA	Latent Dirichlet Allocation
ML	Machine Learning
MOX	Metal-oxide
ROC	Receiver Operating Characteristic curve
SFA	Symbolic Fourier Approximation
SNR	Signal-to-Noise Ratio
TS	time series

1 | Introduction

1.1 Motivation

Today's infrastructure is of critical importance for the reliability and maintenance of modern society, as well as the high degree of globalization. Political or religious conflicts are part of the present time and, unfortunately, under some circumstances, can pose a threat to vital components of current social life. Infrastructure like airports, train stations, and other places of high traffic movements are, regardless of the motives, one of the primary destinations of terrorist attacks. Often times such acts are carried out by placing improvised explosive devices, hidden or disguised, in areas either highly frequented by a large number of people and/or places of most critical nature, like e. g. nuclear power plants (Bennett 2018, pp. 37 sqq.). The detection and appropriate disposal of such explosives is of the utmost importance for the safety of all concerned. Given a scenario for the detection of an unknown or suspect object in such places, determining its risk potential is of the highest urgency. The explosive content of such improvised devices can be found in the form of a powdered, solid, or liquid substance, visually indistinguishable from other harmless substances. Therefore, a method of identifying explosives in a reliable and time-critical fashion is needed.

Several possibilities of detecting explosives exist. One possible distinguishing characteristic is the type of approach used in the following two options: library-based systems, and methods of working in a library-free manner. As an example for a library-based detection technique, sniffer dogs are used very frequently at airports or other places. Due to specific training, the dog is able to identify explosive substances by the sense of smell. Although only for substances that were part of the training, which makes this technique library-based (Konstantynovski 2018, p. 22). The work in this thesis is based on a library-free method instead. In this case, the technique enables the detection of explosives by means of Machine Learning. A training dataset is used to create a statistical model, but unlike a libary-based method, the model is able classify new, unseen data samples by utilizing various algorithms from the area of Data Mining.

A major part of this thesis is based on the work of Konstantynovski *et al.* (n. d.). A significant component of the publication is the contribution to build and improve a specific device, developed as a prototype for the detection of explosives. Equipped with various kinds of sensors, this device is able to make measurements of samples for an unknown material and provide meaningful data to be used for further analysis tasks, i. e. the Machine Learning model. The focus in this thesis is placed on the further development of analytical methods. The hardware setup will be introduced and explained, so that a basic understanding is gained. Advancements regarding the

analytical tasks should open up new fields of application not yet covered by the work done so far regarding this project. Essentially, those tasks consist of two novel use cases, which are the classification of explosives according to their degree of energetic potential; and the identification of explosives, based on their chemical structure. This work will continue to search for alternative ways of working with the provided data to improve upon the existing results.

The advantages of the device being used for the sampling and data-generating procedure are its compactness and modularity, which leads to simple logistical manageability. Given the real-life scenario of a potential security threat by an improvised explosive device or unknown substance, the sampling procedure and evaluation of the data can be done both, in short time and with high degree of certainty regarding the obtained results. In addition to the task of detecting an explosive, the two new use cases provide further insight for the decision process; being able to classify a high-energetic substances and/or identifying subgroups of chemicals, can be of great usefulness for the operator of the device. The development of suitable methods to provide an implementation for both use cases will be the main task of this thesis. The circumstances of working with sensor data, in the context of the detection device, will require additional steps to align with the present method of the detection process.

1.2 Related work

The dataset, as well as some parts of the methodology of Konstantynovski *et al.* (n. d.) have been adapted and extended for further use. Preceding to this, two additional publication exist (Konstantynovski *et al.* 2017 & 2018), with a greater focus on the hardware setup, including the process of development and improvement for the detection device. Regarding the analytical task for the sensor data, in common with Maurer *et al.* (2015), both publications apply Principal Component Analysis for the use case of identifying individual substances. Guaman *et al.* (2019) show a similar approach, but use a combination of Fisher's Linear Discriminant Analysis and Principal Component Analysis. On basis of domain knowledge, with the features extracted from the sensor data, it is possible to identify groups of multiple samples of a substance in the visualization of the first three or two principal components. The method applied in the publications resembles to some extent the framework developed in this thesis, i. e. the dimension reduction on the feature set. However, in the case of the thesis, different ways of implementing and applying the Machine Learning models result in a more flexible and automatic way of working with the data, without being too dependent on domain knowledge.

A major part of the work in this thesis revolves around the extraction of features from sensor data. The method selected for this task was developed and published for the application in the field of time series, i. e. sensor data, by Schäfer (2015b).

Worth mentioning is the method developed by Lin *et al.* (2007), which was published in advance to Schäfer (2015b). While both aim for a symbolic representation of time series data, a crucial difference is the way of approximating the signal. The algorithm applied in this thesis makes use of a Fourier Transform, while Lin *et al.* (2007) calculate mean values instead. An advantage of Schäfer (2015b) is the flexibility in regards to the level of noise reduction. In case of the Fourier Transform, the number of coefficients determine the degree of smoothing, and can be selected freely, while only doing the calculation once. In contrast to the approach of Lin *et al.* (2007), where the calculation for a time series has to be repeated, if the degree of smoothing is changed (Schäfer 2015a, pp. 34 sq.). Both methods have in common that the outcome contains a discrete data structure, representing counts of extracted symbols as the result of the approximation. This opens up new possibilities to apply methods from different fields of Machine Learning, not limited to only Data Mining algorithms for time series.

Many publications exist, utilizing this selected algorithm and performing benchmarks in context of time series classification or similarity search. Some published work exist specifically for the particular combination of two models applied to the results of the feature extraction process in this thesis. The applied models are a dimension reduction technique called Latent Dirichlet Allocation, followed by a clustering approach with K-Means. Twinandilla *et al.* (2018) show an implementation using both algorithms in the field of Natural Language Processing. Here, the performance of detecting significant sentences in online news documents is measured by an external validation score. Similar to the work in this thesis, external metrics are used to determine the optimal number of clusters. Related to this implementation, Bui *et al.* (2017) show an empirical study of various distance measures for the clustering algorithm in combination with the dimension reduction method. Some additional publications can be found regarding the application of the method of Lin *et al.* (2007) and Latent Dirichlet Allocation, as for example in McLaurin *et al.* (2014) or Chen & Qi (2019). In context of the dimension reduction technique and the clustering step applied on the extracted feature set of the method of Schäfer (2015b), no publications of work exist to the knowledge of the author.

1.3 Thesis outline

At first, the theoretical foundation is provided in section 2 for the methods used throughout this work. This includes the application of Machine Learning models in the context of sensor data and the algorithms further applied, to be able to implement a working solution for the specific use cases. Following, section 3 gives an introduction to the hardware of the detection device, which was used to create the dataset, and explains its functionality accordingly. In addition, the necessary data pre-processing steps will

be explained, as well as the methodology used in prior by Konstantynovski *et al.* (n. d.) for the binary classification of explosive and benign materials. The rest of this thesis is structured as follows: section 4 demonstrates the implementation of the new proposed framework; and section 5 presents the results for the application in context of the two new use cases. Finally, section 6 draws a conclusion and provides an outlook for possible future work.

2 | Theoretical foundations

2.1 Chemical principles of explosive materials

Explosive substances appear in various forms as liquid, solid or gaseous. With sufficiently high activation energy, heat and gas can be released abruptly (Köhler *et al.* 2008; as cited in Konstantynovski 2018, p. 16). This energy can be artificially induced. In section 3.1, the procedure to evoke a reaction of explosive materials and gain meaningful data by means of sensor measurements will be described.

For the analysis of the resulting data, there are multiple possible applications. The substance could be classified as explosive/benign by an algorithm trained with positive and negative examples. Another use case would be to investigate the similarity between different substances according to their chemical structure. For this purpose, the explosives can be divided into categories, e. g. inorganic and organic. Inorganic substances are, for example, a combination of Nitrates mixed with fuels like coal powder. Organic substances can mainly be subdivided into Nitrates, Peroxides, Nitratester, and Nitramine (Agrawal & Hodgson 2007; as cited in Konstantynovski 2018, p. 17). Further, explosives can be ranked according to their energetic potential, where especially the detection of very high energetic substances can be of great importance. In the later course of this thesis, the dataset[1] will be used for identifying various groups of substances based on their chemical structure. In addition, explosive substances can be classified by type of use.

2.2 Applying Machine Learning to sensor data

The data processed in this thesis consists of time series (TS), i. e. sensor data, with values observed at specific points in time. Given the structure of such data, conventional statistical methods are not applicable, since independent and identical distributed (*iid*) samples are not available (Shumway & Stoffer 2017, p. 1). The following sections provide the concepts of the methods used in this thesis for analyzing such data.

Repeated observations of a random variable $x_1, x_2, \ldots, x_t$ can be described as a collection $\{x_t\}$, where t is a discrete index. The realization of this stochastic process is called a TS (Shumway & Stoffer 2017, p. 8). In addition, there are also multivariate variants of a TS with r components, denoted as $x_{t1}, x_{t2}, \ldots, x_{tr}$ (Shumway & Stoffer 2017, p. 19). The dataset, described in detail in section 3, consists of multivariate sensor data. For each variable, measurements of $r = 9$ sensors exists. In this case, the consideration of all 9 components is a key factor for every analytical task applied on

[1] A detailed list of all available chemical substances can be found in table A.1 on page 64.

the TS data and needs to be considered. The main task of the thesis is to find ways to work with (multivariate) TS data for the given use cases.

2.2.1 Time series classification

For TS there are multiple scenarios of predicting an outcome based on the data. Forecasting a continuous value like stock prices is seen as often as predicting electricity demand. The particular application in this thesis is not of a continuous outcome but a qualitative instead, making it a classification task. Classifying TS data can be divided into different categories. There are up to six of such subcategories, according to Bagnall *et al.* (2017):

- Whole series: two TS are compared as a vector or by a distance measure
- Intervals: like whole series but reduced to intervals
- Shapelets: finding patterns that are representative for the given class
- Dictionary-based: frequency of recurring patterns, based on histograms
- Combination of the approaches
- Model-based: fitting a generative model to each series and comparing them

The selection of a suitable method in this thesis was determined depending on the produced (data) structure of the generated feature set. An important reason to consider is that datasets containing TS measurements share the property of high dimensionality, because of the number of sample points in a TS. For the reason of applying Machine Learning (ML) algorithms in this thesis, a flexible approach had to be found, allowing for *(i)* robust feature representations of TS data; and be able to *(ii)* reduce the dimensionality effectively, without too much loss in information. The reason for this is that the performance of a ML model is affected by the *Curse of Dimensionality* (Bishop 2006, pp. 34 sqq.). A feature set with $n \ll p$, where the number of observations is much lower than the number of features, can lead to a weak performance and an unnecessary high computational load for the training and prediction task of the model. For this purpose, a dictionary-based approach has been chosen as a feature extraction procedure. This type of algorithm represents a TS through symbols as features, allowing for a broader application of methods, e. g. of the area of Natural Language Processing (section 2.3.1), while still able to parameterize (setting the value for a hyperparameter) and regulate the reduction of dimensionality for the obtained feature set.

As an extension to the six methods presented, Ruiz *et al.* (2021) propose additional frameworks, suited to work with and classify multivariate TS data directly. However in this thesis, the actual part of classifying will be done by a separate method. Since most of the frameworks are too restrictive in the way the classifiers are implemented, they are of no use in this case. The emphasis of selecting an approach is placed on the

extraction procedure of the features themselves and the flexibility to choose and test freely from a set of matching classifiers.

The six categories defined by Bagnall *et al.* (2017) can be broken down even further, depending on the method used. Moreover, it is possible to distinguish between distance-based (Abanda *et al.* 2019) and feature-based (Fulcher & Jones 2014) methods for time series classification. Eventually, even when distance-based methods are applied, the result is or can often times be a feature vector transformed from the original distances, as for example shown by Rohit (2016). The concepts differ predominantly in the way they extract features; for both procedures, the methods used to classify data are largely very similar. As an alternative approach, it is also possible to use a Neural Network architecture for the two tasks of feature extraction and classifying; Hsieh *et al.* (2021) show promising results when working with multivariate TS data. Although for the dataset in this case, the sample size of the available data may be too small for the training of a Deep Learning model.

The work done so far by Konstantynovski *et al.* (n. d.) was feature-based using the obtained feature vectors of each TS for the development of a supervised learning classifier. The extraction process relied heavily on domain knowledge for the most part (further described in section 3.3). The primary task was to use the extracted features for binary classification of the chemical substances. Though it was also tested whether it is possible to utilize the feature set to discriminate even further between different subclasses by applying additional ML methods. Especially the latter will be the main focus of this work and extends the methods applied on the dataset to the present state.

2.2.2 Bag of Symbolic Fourier Approximation Symbols

In this thesis a dictionary-based algorithm was chosen to extract features from sensor data. In essence, those algorithms pass a window of predefined length across the TS to extract subsequences. Every subsequence is then discretized through a symbolic representation. The occurrence of those extracted strings is collected in a histogram, eventually representing each TS through frequencies of strings/words (Bagnall *et al.* 2017). The procedure results in a data structure similar to the Bag of Words (BoW) model, where only the frequency of the words is considered, but not the order of occurrence (Manning *et al.* 2009, p. 117). This opens the possibility for the application of new methods, e. g. one of them being Latent Dirichlet Allocation (LDA) out of the area of Natural Language Processing, further described in section 2.3.1.

Schäfer (2015b) developed a dictionary-based algorithm called Bag of Symbolic Fourier Approximation Symbols (BOSS), which was selected in this thesis for the extraction of features. This procedure makes use of Symbolic Fourier Approximation (SFA) to extract symbols representing patterns in a TS. The SFA approach (Schäfer & Högqvist

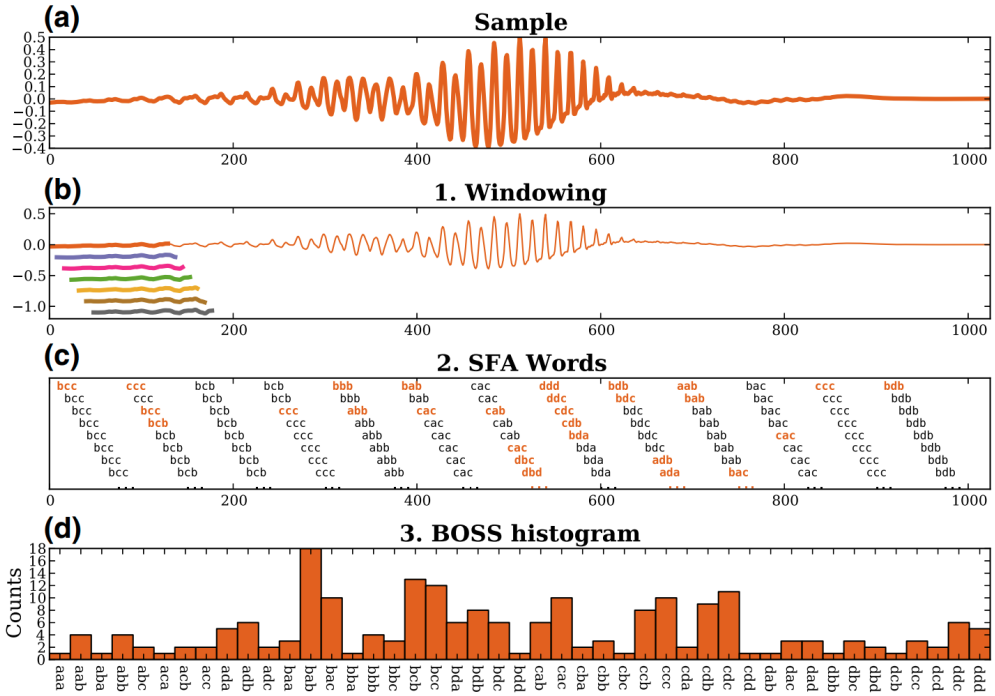


Fig. 2.1: Extraction procedure for the BOSS model on a sample.‡ At the top *(a)*, the sample of a TS is shown for which the histogram *(d)* is being created. In the first step *(b)*, a overlapping sliding window of a set length is passed across the TS, extracting subsequences. For each subsequence *(c)*, a SFA symbol of length 3, consisting of 4 possible characters $\{A_1, B_2, C_3, D_4\}$, is being extracted (marked in orange); consecutive symbols (marked in black) are skipped. The histogram representation *(d)* is built from the frequency of occurrence of those symbols.
‡ Figure taken from Schäfer (2015b).

2012) combines a Discrete Fourier Transform (DFT) for the approximation of the time series, i. e. subsequences, and a discretization step for the extraction of symbols. By applying this concept to the subsequences of the sliding window, the BOSS algorithm is able to build a BoW model. In contrast, the SFA approach extracts only one single symbol for the complete length of a TS. Figure 2.1 shows the procedure of generating a BOSS model on an example of a TS shown in panel *(a)*. In the first step *(b)*, subsequences are being produced for the extraction of SFA symbols. The following step *(c)* applies the SFA procedure to all subsequences. Finally, a histogram *(d)* can be built, representing the TS through the frequency of occurrence of the SFA symbols (Schäfer 2015b). In a next step, numerosity reduction is applied so that „[...] the first occurrence of an SFA word is registered and all duplicates are ignored until a new SFA word is discovered“; this avoids „[...] outweighing stable sections of a signal“ (Schäfer 2015b, p. 10). Visible in fig. 2.1 *(c)*, the symbols excluded by numerosity reduction are not marked in orange.

The scope of approximation of a subsequence by the DFT may be influenced. The number of coefficients of the Fourier Transform can be set accordingly. When applying the BOSS model, the number of the first coefficients is controlled by the hyperparameter

„word length“, which determines the degree of filtering. Choosing only the initial coefficients of the DFT, therefore a small word length, represents the application of a low-pass filter and removes (high-frequent) noise. As a result, the transformed signal is closer to the trend/shape of the original one (Schäfer 2015b). Compared to other procedures, the Fourier transformation was preferred as the approximation mechanism for performance reasons (Schäfer 2015a, p. 39).

The discrete variant of the Fourier Transform (Fourier 1822) can be defined according to Bracewell (2000, p. 260) as

$$F(v) = N^{-1} \sum_{\tau=0}^{N-1} f(\tau)\, e^{-i2\pi(v/N)\tau}$$

where N represents the number of (sample) points of the signal, i. e. subsequence, and $F(v)$ the DFT of the original signal $f(\tau)$. At every $\tau \in \{0, 1, 2, \ldots, N-1\}$ the Fourier Transform is applied, so that for $F(v)$ a tuple $\left\langle \mathbb{R}\mathrm{e}_{v=1,\ldots,N}, \mathbb{I}\mathrm{m}_{v=1,\ldots,N} \right\rangle$ is obtained, normalized by N^{-1}. As an example, the first two coefficients of the DFT are calculated by

$$F(v) = 2^{-1}\left(f(\tau=0) + f(\tau=1)\, e^{-i2\pi(v/2)}\right) \qquad \text{for } v = \{0, 1\}$$

with $N = 2$, resulting in the two tuples $\langle \mathbb{R}\mathrm{e}_{v=0}, \mathbb{I}\mathrm{m}_{v=0} \rangle$ and $\langle \mathbb{R}\mathrm{e}_{v=1}, \mathbb{I}\mathrm{m}_{v=1} \rangle$. In case of the SFA method, the first coefficient $v = 0$ is always discarded, since its only representing the mean of the signal (Schäfer 2015a, p. 40). In the discrete variant of the Fourier Transform, the number of coefficients v is finite and can be limited by the BOSS model with the hyperparameter „word length“, described in more detail below.

Discretizing by mapping the DFT coefficients to symbols is done in a quantization step, after the approximation of the subsequences took place. The quantization assigns coefficients of a continuous space to a discrete one, i. e. symbols (Schäfer & Högqvist 2012). Figure 2.2 shows the procedure called *Multiple Coefficient Binning*, which is used for the quantization of the coefficients in the SFA process. Depending on the word length l set by the BOSS model, there are $l/2$ coefficients of the DFT considered in every subsequence; usually $l \ll n$, where n is constant for the length of the subsequences. For each real and imaginary part of every coefficient, the quantization is applied. By binning the values according to their distribution across all subsequences, a SFA word can be derived for every subsequence. The number of bins depends on the chosen size for the set of symbols $M \in \{N \subseteq \{\mathrm{A}, \mathrm{B}, \ldots, \mathrm{Z}\} \mid \mathrm{A}, \mathrm{B} \in N\}$, and thus, depth of available symbols. At least the first two characters $\{\mathrm{A}, \mathrm{B}\}$ are always required as the minimum value for the parameter $l \geq 2$, else, the creation of bins for the discretization step cannot be done. This is controlled by a hyperparameter of the BOSS model, where a lower alphabet size results in a higher reduction of noise for the original signal and vice versa.

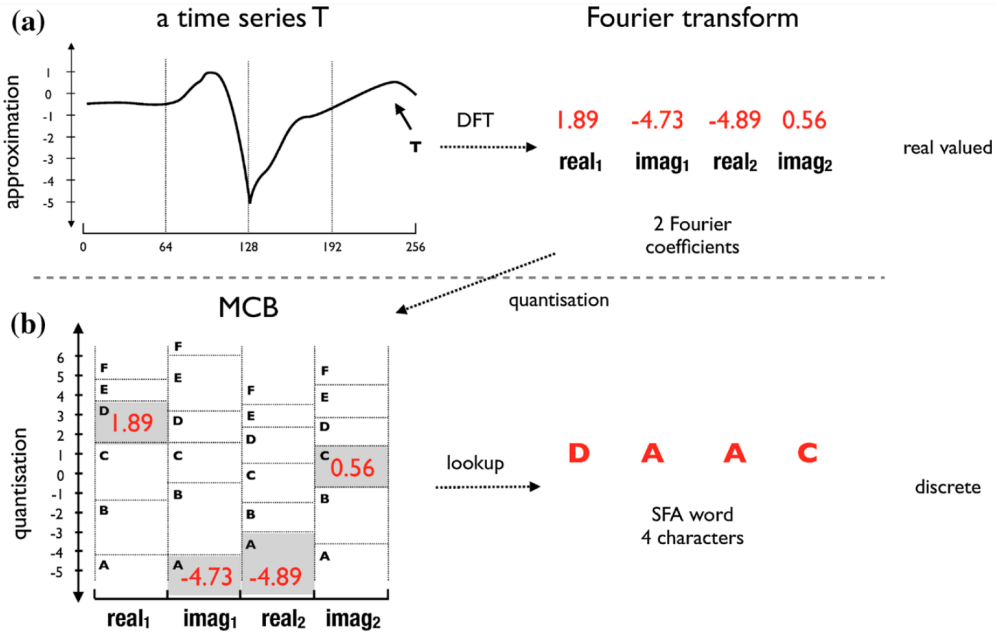


Fig. 2.2: Multiple Coefficient Binning for the quantization of a DFT.‡ In this example, a word length of $l = 4$ is used for *(a)* the subsequence. The method adapts to $l/2$ coefficients for the DFT of all subsequences to calculate *(b)* the width of the bins. For the tuple $\langle \text{real}_i, \text{imag}_i \rangle$ of every i^{th} coefficient, a binning strategy is applied, e. g. same number of points in each bin. After the calculation, all binned coefficients can be mapped to their corresponding symbol, resulting in the final SFA word representation for each subsequence.
‡ Figure taken from Schäfer (2015b).

The quantization process is applied for each subsequence, resulting in multiple words, leading to the BoW representation of the TS (Schäfer 2015b, pp. 5 sqq.).

The computational complexity of the BOSS transformation is $O(n + w \log w)$, where n is the length of a TS and w the length of the sliding window for the subsequences. At the first subsequence, the DFT is calculated with a complexity of $O(w \log w)$. Since there are overlapping coefficients due to the sliding window procedure, all following subsequences can be approximated by the Momentary Fourier Transform instead, which has a complexity of l, the SFA word length; therefore, the complexity is constant for the length n (Schäfer 2015a, p. 106).

The combination of a low-pass filter by using the DFT as an approximation mechanism and the quantization of the subsequences leads to a robust noise-resistant transformation for TS data. The creation of a histogram through the BOSS approach and the representation as a BoW model opens up new possibilities to apply various methods for further data analysis tasks.

2.3 Clustering and dimension reduction techniques

The thesis so far laid down the foundations to extract features from sensor data. The following sections will explain the methods used to build a model based on those features by means of ML. The ML model should be able to generalize well and predict new, unseen TS data. For this task, the methods primarily used in this thesis are built up by two steps:

1.) Preprocessing: in order to reduce the dimensionality of the BOSS feature set by applying LDA.
2.) Unsupervised Learning: clustering the data and leveraging class information for different use cases on the reduced feature set.

Both methods will be described in the following sections. Section 2.3.1 will review the LDA method, which was used for the dimension reduction of the BOSS feature set. Following this, section 2.3.2 gives an introduction to K-Means (KM) for the discovery and extraction of meaningful structures in the data. A short overview of suitable metrics for the evaluation of the clustering performance will be given afterwards.

2.3.1 Latent Dirichlet Allocation

In the area of Natural Language Processing and Text Mining, a typical task for a collection of documents (*corpus*) is to extract semantic topics, which are initially hidden (*latent*), and use them to annotate the documents. The task is predominantly implemented by applying LDA, frequently called *Topic Modeling*. This method is not only limited to Text Mining applications. Several approaches exist in other scientific disciplines, such as medical/biomedical or geographical areas, and software engineering (Jelodar *et al.* 2019).

The assumption behind LDA as a Topic Model is that every document in a corpus can be represented through a distribution of latent topics and each of those topics is defined by a distribution of words of the documents in the corpus. In essence, LDA is an unsupervised learning approach and among other things, can be used as a dimension reduction technique (Jelodar *et al.* 2019). Figure 2.3 visualizes the LDA model introduced by Blei *et al.* (2003) to find K topics in a collection of documents D. The process consists of several latent random variables and one observable, the word count $w_{d,n}$ for a document d of a word n, while the latent variables need to be inferred. The latter are β_k as a $K \times V$ matrix for the distribution of words over K topics, where V is the size of the lexicon (number of words); the distribution θ_d of topics over documents; the single topic assignment $z_{d,n}$ of a word n in a document d.

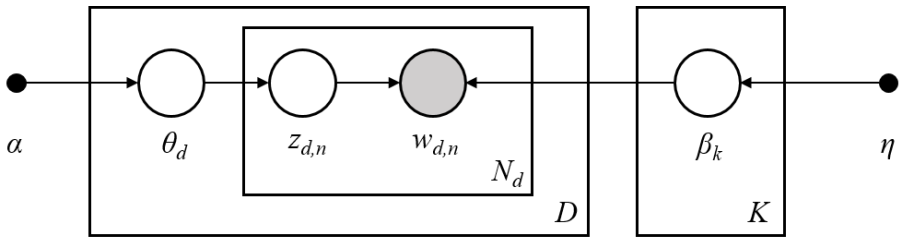


Fig. 2.3: Graphical plate notation for the LDA model.‡ Every round node is a latent random variable, while the gray one $w_{d,n}$ being the exception for the observable word count of a document. The arrows show dependencies between variables. The rectangles represent the corpus D of all documents, a specific document N_d in the corpus, and K topics. The LDA model can be adjusted by two priors α, η, and a hyperparameter K for the number of topics.
‡ Figure taken from Hoffman *et al.* (2013).

The LDA model is a generative process with repeated sampling for every variable visible inside a rectangle box in fig. 2.3. The particular process can be described as follows:

1.) Draw $\beta_k \sim \mathcal{D}(\eta)$ for each topic $k \in \{1, \ldots, K\}$.
2.) For each document $d \in 1, \ldots, D$:
   a) Draw topic proportions $\theta_d \sim \mathcal{D}(\alpha)$.
   b) For each word position $n \in \{1, \ldots, N\}$:
      i) Draw topic assignment $z_{d,n} \sim \text{Multinomial}(\theta_d)$.
      ii) Draw word $w_{d,n} \sim \text{Multinomial}\big(\beta_{z_{d,n}}\big)$.

The priors for the Dirichlet distribution η and α influence its shape. A higher value represents a more uniform distribution of topics/words, and a lower a more sparse one. To estimate the parameters of the model, the exact inference for the posterior distribution $p(z, \theta, \beta \mid w, \alpha, \eta)$ has to be calculated with

$$\frac{p(z, \theta, \beta \mid \alpha, \eta)}{p(w \mid \alpha, \eta)} = \prod_{i=1}^{K} p(\beta_i \mid \eta) \prod_{d=1}^{D} p(\theta_d \mid \alpha) \left(\prod_{n=1}^{N} p\big(z_{d,n} \mid \theta_d\big)\, p\big(w_{d,n} \mid \beta_{1:K}, z_{d,n}\big) \right)$$

This step is usually approximated with e. g. Variational Inference $q(z, \theta, \beta \mid \lambda, \phi, \gamma)$, where instead substitutions are used for the different distributions. This allows for a simpler calculation, since the posterior is computationally intensive to solve. By using the Kullback-Leibler divergence, the similarity between the original distribution and its substitute can be approximated (Hoffman *et al.* 2013). Furthermore, the model shown in fig. 2.3 is a smoothed variant of LDA by using the prior η. This prevents words that do not occur in the corpus at the time of the creation of the model from having a probability of zero in new, unseen documents. A precondition of LDA is the representation of documents as discrete data, i. e. the BoW model. The sensor data in

this thesis, transformed by the BOSS method, can be used for this approach. Section 4 will show the implementation of both methods combined.

2.3.2 K-Means

The purpose of clustering data can be versatile. Different applications occur of which according to Jain (2010), the following main categories can be defined:

- Underlying structure: gain insights (anomalies, hypotheses, identify features)
- Natural classification: identify similarity among organisms
- Compression: organizing and summarizing data

The use case in this thesis is to gain insights of the underlying structure, specifically, to cluster the feature set obtained through the application of LDA on the BOSS representation of the sensor data.

The goal of clustering is to divide a collection of n observations with p features into disjoint groups. A partition $C_1, \ldots, C_K$ needs to be found with

$$C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$$

and

$$C_k \cap C_{k'} = \emptyset \text{ for all } k \neq k'$$

where each observation has to be a member of only one cluster (James *et al.* 2013, pp. 385 sq.). All clustering algorithms, regardless of the specific method selected, aim for similarity/isolation within the clusters, while achieving high discriminatory power between them. Different metrics exist to measure the performance of the clustering results; section 2.3.3 will give an overview of the metrics used for the evaluation of the models built in the thesis. This section will primary focus on the introduction of the KM algorithm and its variation K-Means++ as the selected clustering method.

In general, there are two major groups of clustering algorithms, hierarchical and partitional. The KM algorithm is a partitional clustering procedure. In contrast to the hierarchical method, all clusters are searched for simultaneously, while the clusters in a hierarchical model are found successively instead (Jain 2010). In the first step of the KM algorithm, the hyperparameter for the number of clusters K is chosen. The goal of KM is to minimize the within-cluster sum of squares

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} || \, x - \mu_i \, ||^2$$

for all x data points according to the related cluster mean μ_i. The standard naïve KM implementation (Lloyd 1982) proceeds with assigning K data points called *centroids*

randomly in the dataset, representing the initial cluster centers. In a next step, each one of the remaining data points is being assigned to its optimum centroid. This is done by calculating the squared Euclidean distance for every data point to each k centroid with

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2 \forall j, 1 \leq j \leq k \right\}$$

where x_p is a data point, S_i a cluster, and m_i and m_j are corresponding centroids. This process is iterative, after each iteration t all centroids are re-assigned again. The initial centroids will be replaced by the mean value of the respective cluster for $t + 1$ with

$$m_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j$$

After the new centroids have been calculated, the assignment of data points to centroids/clusters repeats itself. The repetitions are done until the maximum number of iterations is reached or no more changes occur (Aggarwal & Reddy 2014, pp. 89 sqq.).

One disadvantage of naïve KM is the random assignment of the initial cluster centroids. At worst, the starting positions may heavily influence the convergence to an optimum as well as the quality of the result. Arthur & Vassilvitskii (2007) present an alternative approach called *K*-Means++, which makes use of *careful seeding* for the initialization phase. Instead of setting all centroids at once, the allocation takes place one after the other. Figure 2.4 shows an example illustration of the initialization phase. After the first centroid has been chosen randomly, the next centroids will be selected based on the highest probability. The probability for each point is calculated with a D^2-weighting, which is defined by

$$\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$$

where for every data point x the distance to its nearest centroid will be calculated and weighted by the sum of all distances. Through sampling of the obtained probabilities, the procedure applies an interpolation step, which has the advantage of being more resistant against outliers in the dataset. In addition, Arthur & Vassilvitskii (2007) show an advancement compared to the standard KM algorithm in terms of time to convergence, despite the effort of calculation for the initial allocation of the centroids. The computational complexity of KM++ is lower with $O(\log k)$, by comparsion, KM has $O(n\,k\,t)$, whereby k is the number of clusters, n the number of observations, and t the (maximum) number of possible iterations.

Choosing the number of clusters K for the algorithm can depend on several factors. Section 4.3 will describe the method used to determine the optimal hyperparameter

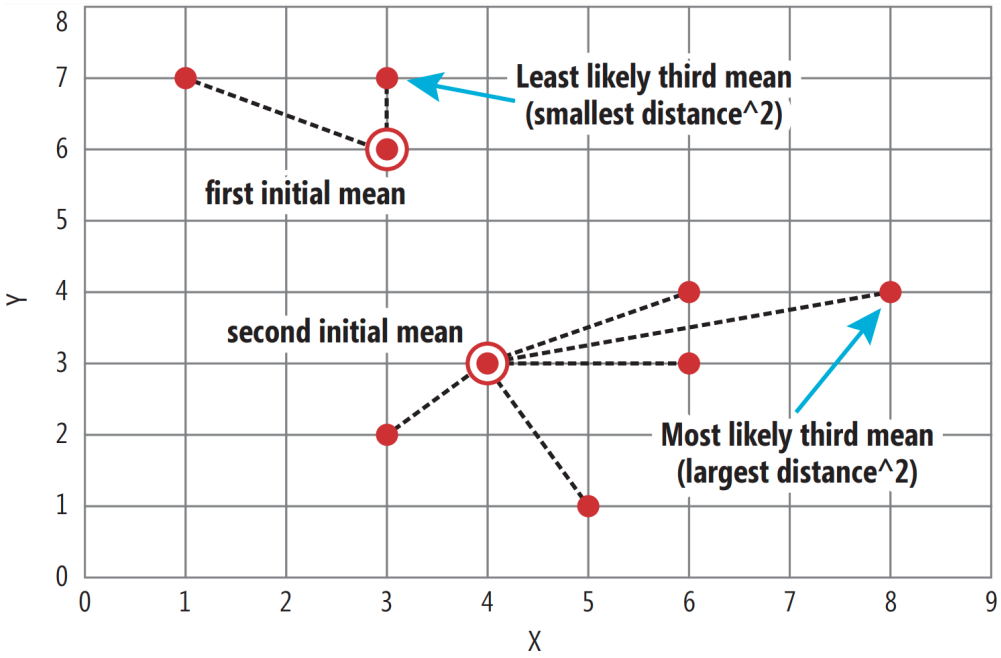


Fig. 2.4: Illustration of the KM++ initialization phase.[‡] In this example, two of the total of $K = 3$ initial centroids have already been found, and the last one still has to be determined. Calculating and sampling the D^2-weightings for the remaining data points results in point $\langle 8, 4 \rangle$ with the highest probability as the third missing centroid.
[‡] Figure taken from McCaffrey (2015).

setting given specific use cases. The KM++ algorithm forms the last piece of the ML pipeline used in this thesis.

2.3.3 Clustering performance metrics

So far, the KM algorithm and its improved variant have been explained, as well as the purpose of clustering in general. Given the nature of cluster methods in an unsupervised learning setting, the parametrization and validation of the model normally does not rely on known conditions, e. g. class labels, in contrast to supervised learning (Hastie *et al.* 2009, pp. 485 sq.). Therefore, either domain knowledge needs to be applied, targeted to solve the specific use case, or some kind of performance evaluation can be considered, aiming to compare different modeling approaches. Various metrics exist, which enable a quality comparison of the results produced by the clustering model. This section will present a selection of scores to rank and compare different models, to be able to choose the best fitting one on basis of the metric evaluation.

The performance of a clustering result can be determined by „[...] procedures that evaluate the results of cluster analysis in a quantitative and objective fashion“ (Jain & Dubes 1988, p. 143). Here, a distinction can be made between metrics that evaluate the structure of a single clustering partition and those that can make use of class labels for the calculation of the score; these are also called internal and external validation

measures, respectively (Tan *et al.* 2019, p. 571). According to Tan *et al.* (2019, p. 571), different goals can be achieved when evaluating clusters:

- Clustering tendency: non-random structure in the data
- Correct number of clusters
- Goodness of fit without external information
- Goodness of fit with external class labels
- Comparing two sets of clusters

Whereby 1–3 are strictly internal measures, while item four exclusively makes use of external information, and the last one can be applied in both cases. The focus in this thesis is placed on measures using external information, for the reason that class labels for different use cases are already available.

A major difference is that internal *indices* (validation measures) measure the cluster cohesion (compactness, tightness) and cluster separation (isolation). In contrast, the external indices compare the clustering partition to an already existing (external) one (Tan *et al.* 2019, p. 572). The possibility of using external indices as a validation measure has the advantage to obtain „[...] clustering results which can match the categorization performance by human experts“ (Wu *et al.* 2009); this is one of the main goals of the thesis. A wide range of metrics exists, of which an overview of the selection used will be given.

Four different external indices have been applied to the clustering results. By using multiple metric scores without relying on a single one, a robust ranking can be implemented, which will be explained in more detail in Section 4. The indices chosen are:

- Rand index (adjusted)
- Mutual Information (adjusted)
- V-Measure
- Fowlkes-Mallows index

The first two measures are one of the most widely used when working with known class labels. Further, both are adjusted for chance, which leads to the following two advantageous properties: *(i)* constant value of „0“ if the partitions (clusters & class labels) are independent (random); and in return *(ii)* constant value of „1“ for identical partitions (Romano *et al.* 2016).

Adjusted Rand index. Introduced by Hubert & Arabie (1985) as an advanced approach to Rand's index (Rand 1971). According to Kuncheva & Hadjitodorov (2004), the adjusted index can be defined by

$$\mathrm{ARI}(A,B) = \frac{\sum_{i=1}^{c_A}\sum_{j=1}^{c_B}\binom{N_{ij}}{2} - t_3}{\frac{1}{2}(t_1+t_2) - t_3} \in [-1,1]$$

where a maximum score of „1“ indicates a perfect clustering result, pure random labeling is indicated by a score of „0“, and „−1“ indicates non-random patterns worse than the random one. The index is calculated with

$$t_1 = \sum_{i=1}^{c_A} \binom{N_i.}{2}, \qquad t_2 = \sum_{j=1}^{c_B} \binom{N_{.j}}{2}, \qquad t_3 = \frac{2\,t_1\,t_2}{N(N-1)}$$

Two partitions, A and B, are compared. The number of observations in all clusters is N_i for partition A and N_j for partition B. The number of clusters in each partition is denoted as c_A and c_B. An assumption is made, that both partitions are drawn randomly, in which case the value would be at constant 0. If both partitions are not clustered independently but identical, the value would be at 1.

Adjusted Mutual Information. Proposed by Vinh *et al.* (2010) as the Mutual Information between two partitions U and V, adjusted by chance, defined as

$$\mathrm{AMI(U,V)} = \frac{I(\mathrm{U},\mathrm{V}) - \mathrm{E}\{I(\mathrm{U},\mathrm{V})\}}{\max\{H(\mathrm{U}),\mathrm{H(V)}\} - \mathrm{E}\{I(\mathrm{U},\mathrm{V})\}} \in [0,1]$$

where both partitions are identical with a score of „1“, and independent/random with a score of zero. The AMI is calculated with

$$H(U) = -\sum_{i=1}^{R} \frac{a_i}{N} \log \frac{a_i}{N}, \qquad I(U,V) = \sum_{i=1}^{R}\sum_{j=1}^{C} \frac{n_{ij}}{N} \log \frac{n_{ij}/N}{a_i\,b_i/N^2}$$

where H is the entropy of a partition, and I the mutual information between two partitions. Being an adjusted index, the result value shows the same pattern with a constant value for identical or independent clusters.

V-Measure. Published in Rosenberg & Hirschberg (2007) to describe the validity of a clustering result. The V-Measure is the harmonic mean of two additional objectives, *homogeneity* and *completeness*. Both have a different desirability regarding the outcome of the clustering: homogeneity scores high if the clusters are pure, i. e. each cluster with observations of the same class only; the completeness measure aims for all members of a class in the same clusters, independent of other observation from different classes in that cluster. The V-Measure as the harmonic mean of both, represents the trade-off between a clustering partition with every observation gathered into one large cluster (high completeness), and a partition where all observations are exclusively in their own cluster (high homogeneity). The definitions for homogeneity h and completeness c are

as follows

$$h = \begin{cases} 1, & \text{if } H(C,K) = 0 \\ 1 - \frac{H(C \mid K)}{H(C)}, & \text{else.} \end{cases}, \qquad c = \begin{cases} 1, & \text{if } H(K,C) = 0 \\ 1 - \frac{H(K \mid C)}{H(K)}, & \text{else.} \end{cases}$$

where $H(C,K) = 0$, with K being the cluster partition and C the class labels, is the optimal case for the homogeneity score. A result of zero for the entropy represents pure clusters. Otherwise, the value gets normalized by the maximum reduction of entropy $H(C)$. In contrast, the completeness score is interpreted as the opposite of the homogeneity value. The V-Measure is thus defined by

$$V_\beta = \frac{(1+\beta)\, h\, c}{(\beta\, h) + c} \in [0,1]$$

which is the harmonic mean of both, homogeneity and completeness, with a weighting factor β. For $\beta > 1$, completeness has a greater impact, and homogeneity for $\beta < 1$. The value range of the score is similar to the one of Mutual Information.

Fowlkes-Mallows index. Introduced by Fowlkes & Mallows (1983) can be defined according to Halkidi *et al.* (2001) as

$$\mathrm{FM} = \frac{a}{\sqrt{m_1\, m_2}} \in [0,1]$$

where the value range of the score is similar to the one of Mutual Information too. The index is calculated with

$$m_1 = \frac{a}{(a+b)}, \qquad m_2 = \frac{a}{(a+c)}$$

where a is the number of observations with identical cluster memberships in both partitions, b the number of observations in an identical cluster in one but not the other partition, and c as the opposite of b.

For calculating clustering metrics, often times a contingency matrix can be helpful. The matrix provides a quick evaluation of the results obtained through a clustering procedure. Table 2.1 shows an example of a contingency matrix. Given a dataset, two partitions are present. The first one P through the application of a clustering algorithm, containing the clusters, and the other as a known ground truth C with the class labels to compare against. In P, the number of obtained clusters is denoted as K, and for C the number of classes as K'. Each cluster is shown with its number of

Tab. 2.1: A contingency matrix for the evaluation of clustering results.‡ The table shows two partitions, P produced by a clustering algorithm, and C as the known ground truth containing the class labels. The number of clusters and classes is denoted as K and K', respectively. The number of objects in each cluster P_i for every class C_j is denoted as n_{ij}. This allows to read the overlap between both partitions.
‡ Own table based on Wu *et al.* (2009).

	Partition C				
Partition P	C_1	C_2	$\dots$	$C_{K'}$	$\sum$
P_1	n_{11}	n_{12}	$\dots$	$n_{1K'}$	$n_{1.}$
P_2	n_{21}	n_{22}	$\dots$	$n_{2K'}$	$n_{2.}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
P_K	n_{K1}	n_{K2}	$\dots$	$n_{KK'}$	$n_{K.}$
$\sum$	$n_{.1}$	$n_{.2}$	$\dots$	$n_{.K'}$	n

contained observations for every P_i cluster and each class C_j as n_{ij} (Wu *et al.* 2009). This allows for a simple display of overlap between both partitions.

The section provided an overview of cluster evaluation methods, which are being used in this thesis. Based on the scores provided, the models can be ranked and the best one picked according to its performance. The class labels used for the external validation measures depend on certain use cases, presented later on in section 4.

3 | Dataset

This section uses the publication of Konstantynovski *et al.* (n. d.) as the single source of explanation, if not stated otherwise. This specifically includes everything regarding the hardware setup and the description of different data analysis methods for the experiments. The data worked with in this thesis was made available by DLR-PI[1] („German Aerospace Center“).

3.1 Experiment design and device setup

To get an understanding of the data-generating process, it is first necessary to show the procedure of sampling and collecting data with the device. The experiment consists of specific steps, which are identical for every sample that is being tested. These steps follow a fixed time schedule which influences the structure of the data, described in later sections. Therefore at first, the course of the experiment will be explained accordingly. For the hardware perspective, a quick introduction to the device setup and its sensors will be shown afterwards.

3.1.1 Data-generating process

Given a substance of unknown nature, hereafter also referred to as *analyte*, several ways of determining its risk value, e. g. danger of explosion, can be taken. For the device used in this case, a standardized procedure is applied, which ensures consistent data quality across all experiments.

For every analyte, multiple samples are taken. One individual sample is a dissected subset of the analyte. Every sample will be part of the decision process for the analyte. The process can be broken down into four essential steps:

1.) take N samples of an analyte
2.) run the experiment with the device for every sample $x_n \in N$
3.) apply a predictive model, e. g. binary classifier, on the data obtained from each sample
4.) the majority vote of all samples determines the outcome (explosive/benign) for the analyte

As an example for this binary classification scenario: an unknown substance would be declared as explosive, if at least 6 out of 10 samples are detected as positive/explosive by the predictive model. The prediction takes place in an offline setting, after the experiments ended and the data has been saved persistently to disk.

[1] Deutsches Zentrum für Luft- und Raumfahrt e. V., Institute for the Protection of Terrestrial Infrastructures.

An important assumption is that measurement errors, be it by the operator or the device itself, cannot be fully excluded. For this reason, it is required to do the sampling of the analyte multiple times, e. g. from 3 up to 10 times. This ensures an error reduction if in one or some of the samples sensor errors occur during the experiment or different types of hardware and/or software faults happen. A single sample measurement of an analyte is also referred to as a *run*. For every analyte, each run takes place independently and successively, one after the other, with the important fact that the samples originate from the same single analyte/substance. This leads to

$$x_1, x_2, \ldots, x_n \in \mathbb{R}^9$$

where every run x_i includes measurements from nine sensors. Hence, each run has a multivariate property.

To be able to make measurements of a sample, some sort of chemical reaction needs to be triggered. For this purpose, the sample gets heated in a specific chamber of the device. The reaction to the thermal activation can then be measured with the sensor setup. A custom software on the device controls all steps needed to start and end a run. Table 3.1 shows the time sequence for an experiment with the device. In essence, the schedule can be broken down to three sections, where each section consists of different steps. In the first section within the interval of [0, 119] seconds, the device performs a self-test, checking the sensors and other critical system components.

Tab. 3.1: Time sequence used to perform an experiment run (time in seconds). In the first section up to step 5, a self test for the system is being done. Within the second part of step 6–8, the actual measurement of the sample takes place. The last two steps start a self-cleaning procedure.

Step nr.	Time [s]	Description	Cumulative time [s]
1	10	Flushing the reaction chamber	10
2	30	Flushing the gas sensor chamber	40
3	1	Sampling blank	41
4	30	Measuring blank sample	71
5	48	Flushing the gas sensor chamber	119
6	12	Thermal activation of the sample, homogenization of the gas phase	131
7	1	Sampling	132
8	30	Measuring the sample	162
9	360	Flushing both chambers	522
10	60	Flushing the gas sensor chamber	582

During (119, 162] seconds, the thermal activation of the sample takes place. Especially this section is of most use for the data analysis part further described in section 3.3. In (162, 582] seconds, the last two steps are being used to clean the chambers and pipes.

3.1.2 Description of the hardware

The device which has been used for the experiments can be seen in fig. 3.1. In the reaction chamber the thermal activation of the sample takes place. Attached to this chamber are three physical sensors: pressure, and the photodiodes for infrared and ultraviolet light. The emitted gas flows via pipes to another chamber, where Metal-oxide (MOX) gas sensors are located. The MOX sensors are a particular subcategory, which are able to detect even small amounts of gases, and are especially inexpensive to produce (Konstantynovski 2018, p. 1). This kind of setup comes with a high degree of modularity and extensibility; being able to quickly replace defective sensors or other hardware components allows for a robust operational capability.

The complete sensor setup can be found in table 3.2. In the case of gas sensors, multiple manufacturer models have been deployed. Not every sensor responds to the same type of gas. For example S7, S8 and S9 are able to detect Nitrogen Oxide, while S5 responds to Methane (Konstantynovski 2018, p. 67). Given the circumstances of an

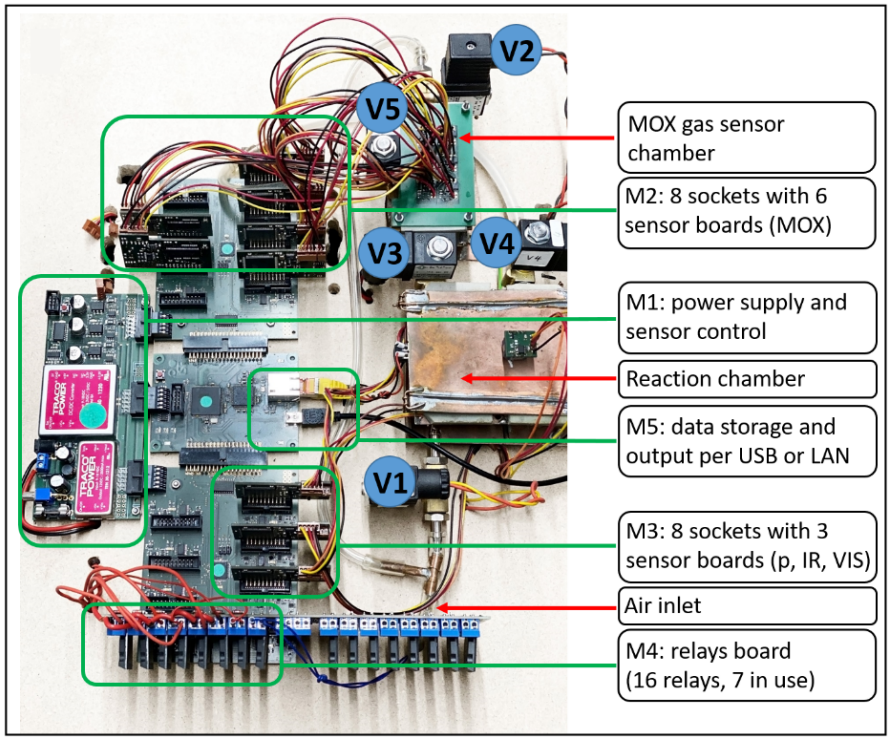


Fig. 3.1: Photographic view of the device electronics.[‡] Physical sensors and gas sensors are separated from each other. While the latter comes with its own chamber, the physical sensors are attached directly to the reaction chamber.
[‡] Figure taken from Konstantynovski *et al.* (n. d.).

Tab. 3.2: Different sensors which are built into the detection device. The first three sensors measure physical activities (pressure and light) and are attached to the reaction chamber, in which the thermal activation of the sample takes place. The remaining six gas sensors are located in a separate chamber and measure the concentration of different types of gases in the air.

ID	Type	Target	Name
S1	light	ultraviolet	BPW21R
S2	pressure	excess pressure	AK2
S3	light	infrared	PT511-2
S4	gas	NO_x, O_3	UST-5333
S5	gas	CH_4	AS-MLK
S6	gas	NO_x	UST-Kosta 6
S7	gas	NO_x	UST-Cologne
S8	gas	NO_x	AS-MLN
S9	gas	NO_x	UST-7333

unknown substance, the combination of different gas sensors is necessary to achieve the broadest possible coverage.

Regarding the time sequence schedule shown in table 3.1, physical and gas sensors differ in the length of the recorded measurements. While gas sensors output data over the whole time frame of the experiment (582 seconds), physical ones only measure in the interval $[119, 122)$ seconds, starting from step 6 onward for 3 seconds, stopped shortly after the thermal activation of the sample takes place. Additionally, both sensor types operate at different frequencies. Physical sensors record with 10.000 Hz and gas sensors with 500 Hz. This leads to different time lengths, which needs to be considered for the data preparation steps described further in section 3.2.2.

3.2 Overview of available data

In total, 27 analytes were sampled with the device, of which 21 are explosives and 6 are benign substances.[1] This data was used by Konstantynovski *et al.* (n. d.) for the development of a decision-making algorithm in a supervised learning setting. The algorithm is able to distinguish between harmful explosive substances and benign ones with a very high detection rate. For this classification task, a train and a test set have been used. The train set consists of 13 explosive and 4 benign analytes. For each of the substances 10 runs have been conducted. In contrast, the test set consists of 8 explosives and 2 benign analytes with only 3 runs for each substance.

[1] A detailed listing of all analytes can be found in table A.1 on page 64.

3.2.1 Dataset structure

The data obtained through the experiments consists of sensor responses in form of multiple TS. For every single one of the nine sensor responses, a TS gets written to a table. Each table holds one run. Given the experiment size of 27 analytes, there are in total 170 runs for the training set and 30 for the test set. The test set holds up to 270 single TS and the training set up to 1.530. The data in its raw version can be seen in table 3.3. Combining measurements for gas and physical sensors in the same table

Tab. 3.3: An example of the raw data of the sensor responses. The time duration differs from physical ($t_{1,2,3}$) to gas ($t_{4,\ldots,9}$) sensors. Therefore, missing values („n. a.“) occur for physical sensors. Furthermore, the timestamps for the start of the recordings are not equal between the two groups.

$\text{Time}_{t=1}$	$\text{Sensor}_{t=1}$	$\text{Time}_{t=2}$	$\text{Sensor}_{t=2}$	...	$\text{Time}_{t=9}$	$\text{Sensor}_{t=9}$	Run	Analyte
119,0000	8831	119,0000	40146	...	0,000	23746	1	X
119,0001	8823	119,0001	40104	...	0,002	23832	1	X
⋮	⋮	⋮	⋮		⋮	⋮	⋮	⋮
121,9999	9437	121,9999	41501	...	59,998	23780	1	X
n. a.	n. a.	n. a.	n. a.	...	60,000	23773	1	X
⋮	⋮	⋮	⋮		⋮	⋮	⋮	⋮

schema leads to inconsistent results. The sensor response of the physical sensors gets recorded in the time frame of $[119, 122)$ seconds instead of 0–582 seconds. In addition, the frequency of 10.000 Hz is higher by a factor of 20. This results in different lengths for the TS, which makes a new data format necessary before continuing to work with the data.

3.2.2 Data preparation

This section will illustrate the process of transforming the raw data into a more usable format. The steps explained in particular are:

1.) New data layout
2.) Interpolation of outliers
3.) Downsampling

Every single one of the three steps are fundamental when working with the dataset and will be explained briefly.

To work with the data, the tables need to be converted first. Therefore, a single, uniform time index column needs to be provided to avoid redundancy. The TS should either be stored in a wide or long table format to achieve a tidy data format. This

simplifies the handling for following steps, like data analysis or the extraction of features (Wickham 2014). The wide format prefers rows with (sensor) values of the same point in time. This requires a split between the two groups of physical and gas sensors into separate tables, because of different time index values. The result would be identical to table 3.3, only with a single time column instead, and two resulting files (S1–S3 & S4–S9). Alternatively, the long format can be used, of which an example can be seen in table 3.4. This allows to store TS of arbitrary length in one table. Using this

Tab. 3.4: Long data format for the TS measurements. This data format allows to store TS of arbitrary length in one table.

Analyte	Run	Sensor	Time	Value
X	1	S1	119,0000	8831
X	1	S1	119,0001	8823
⋮	⋮	⋮	⋮	⋮
X	1	S9	582,0000	24654

format has the advantage of storing both sensor types in the same table schema, as opposed to two different files.

When running experiments with the device, some sort of signal disturbance can occur, which may manifest itself through (unwanted) noise. The reasons for this are diverse. It is important to clean the sensor signal of any disturbance, as good as possible. By doing so, the valid part of the sensor response should not be affected by the procedure, but the corrupt parts should be diminished effectively. Cleaning the signal of unwanted noise will be especially important for the extraction of features or other methods applied to the data (Reimann & Schütze 2013, pp. 84 sq.). As an example, fig. 3.2 is showing the same signal before and after the removal of an interference. The plot on the top shows a disturbance in the sensor response at the very beginning. On the bottom, the pattern has been removed and is not visible anymore. To detect those kind of outliers in the signal, the Grubbs Test (Grubbs 1950) is applied to subsections of the sensor response. The two-sided test is used to detect outliers in both directions and it is defined by

$$G > \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\alpha/(2N),\,N-2}}{N-2+t^2_{\alpha/(2N),\,N-2}}}$$

with a significance level of $\alpha = 0{,}05$. The statistical test needs to be applied to subsections of the signal, which have to be large enough to detect valid outliers, but not chosen too large, so that no outliers are found anymore by the test. The sample size was set to $N = 500$ data points, after trying various values and inspecting the quality of the results. Given the significance level α and the sample size N, the procedure is applied

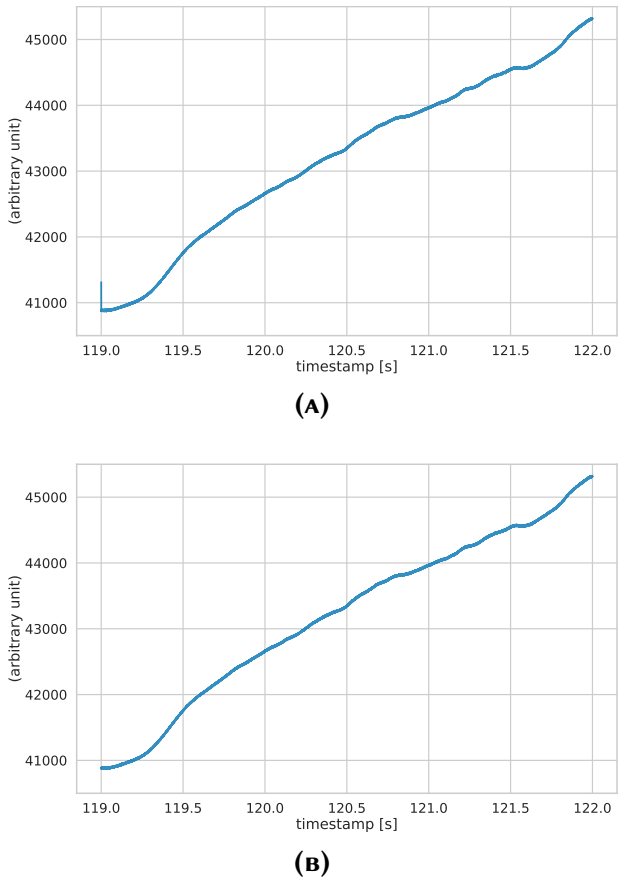


Fig. 3.2: An outlier caused by interference in the sensor response (arbitrary unit on y-axis). Both plots showing the sensor response of the pressure sensor for the fifth run of the analyte Geosit. The first one *(A)* shows a signal interference right at the start at 119 seconds, causing an unusual high peak in the recorded measurements. In the plot on the bottom the pattern is not visible anymore, it has been removed by applying an interpolation procedure.

to the complete TS. For this purpose, a sliding window was used, which increments by 50 indices. Starting from the beginning $\{x_1, \ldots, x_{500}\}$, to $\{x_{51}, \ldots, x_{550}\}$ until the end of the TS. Every identified outlier will be replaced by interpolating the value according to

$$\frac{|\, x_{i-1} - x_{i+1} \,|}{2} + \min(x_{i-1}, x_{i+1})$$

where x_i is the given outlier. This is a pre-processing step that should be applied to the data prior to any other step, since single or multiple outliers could distort the application of the averaging methods followed next (Reimann & Schütze 2013, p. 84).

After the removal of outliers, all TS undergo a reduction of the sampling rate, i. e. change of frequency. This will allow faster processing times and a further reduction of remaining noise as a result. The procedure for the reduction of the sampling rate is split up according to the two sensor types. For physical sensors, the original frequency rate of 10.000 Hz will be reduced to 500 Hz; for gas sensors, the sampling rate gets reduced from 500 Hz down to 2 Hz. This is done by aggregating data points in

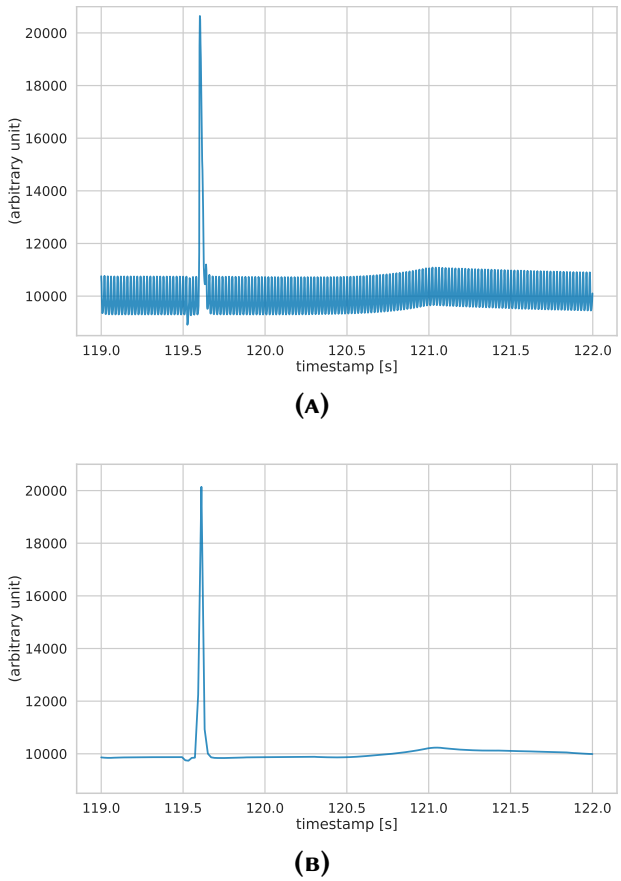


Fig. 3.3: Downsampling of a sensor response (arbitrary unit on y-axis). The plots show a signal of the UV sensor (S1) for the first sample of the analyte HMTD. On the bottom *(B)* the downsampling procedure has been applied.

a moving window for the full length of the TS, similar to the step of identifying outliers, although without overlapping windows. The arithmetic mean will be calculated for each window as an aggregation function. The targeted sample rate can be reached by setting the window size accordingly. For gas sensors, a window size of $N = 250$ will result in a sampling frequency of 2 Hz. The window size for the physical sensors needs to be set to $N = 20$ for a reduction of the sampling rate down to 500 Hz. After the reduction step, the length of the TS for physical sensors went down to 1.500 values from 30.000. Likewise, the data points for the measurements from gas sensors were reduced from 291.000 to 1.164. Figure 3.3 visualizes an example of the downsampling procedure. The top panel *(A)* shows the signal of the UV sensor (S1) for the first run of the analyte HMTD. On the bottom *(B)*, the downsampled version is visible.

3.2.3 Sensor response

To visualize the data produced by the experiments, each sensor response[1] of a run can be plotted as a single TS. Figure 3.4 gives an example for the gas sensor UST-7333 (S9). The plot shows the measurement for the first run of the analyte Pikramid. The orange dashed lines represent the transition between all three main sections of the experiment (table 3.1). In this example, shortly after the thermal activation, a clean response from the sensor to the gas emissions can be seen at around 130 seconds. Also noticeable is the strong increase of the signal showing at around 520 seconds, at the time when the self-cleaning procedure starts and the chambers are getting flushed. This indicates the recovery of the sensor back to its usual state.

An example for the sensor response given by physical sensors can be seen in fig. 3.5. The plots show the response of the UV light *(A)* and pressure *(B)* sensor to a sample from the analyte PETN. The strong increase in the signal happening shortly after 120,5 seconds indicates a clear reaction of the sample to the thermal activation. For the pressure sensor, a decrease after the signal peak can be observed. In contrast to this, the UV sensor shows a smaller, second peak at around 121 seconds. This is due to the remaining glow of the heater in the reaction chamber.

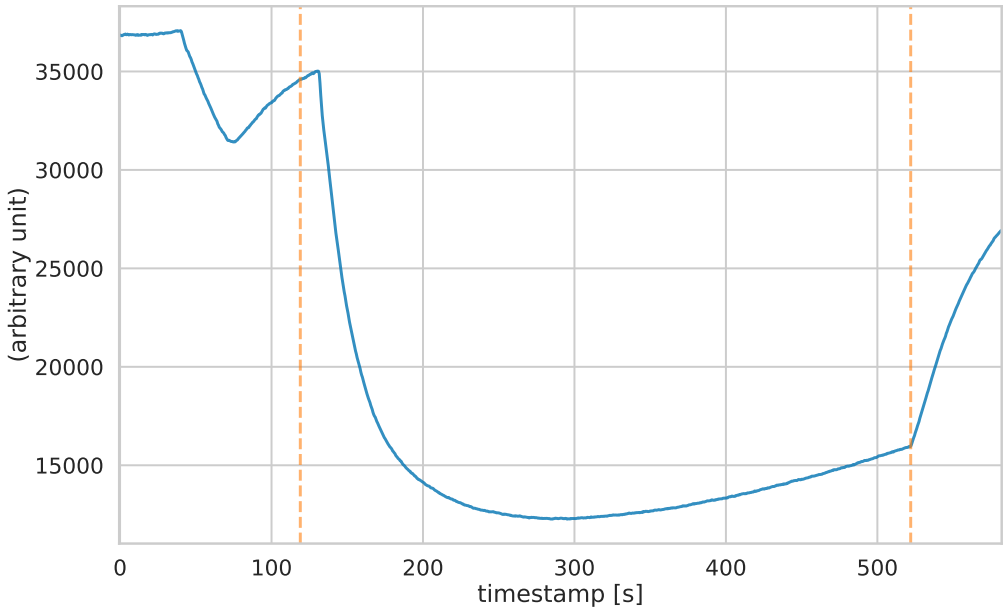


Fig. 3.4: A typical response from a gas sensor. The plot shows the measurement of UST-7333 (S9) for the first run of the analyte Pikramid. Marked in orange are the transitions between the three different sections of the time sequence schedule. The response to the heating of the sample, as well as the recovery of the sensor is visible at around 130 and 520 seconds, respectively.

[1] All sensor values (y-axis) shown in this thesis are without unit and on an arbitrary scale, based on the electronic hardware circuit of the device.

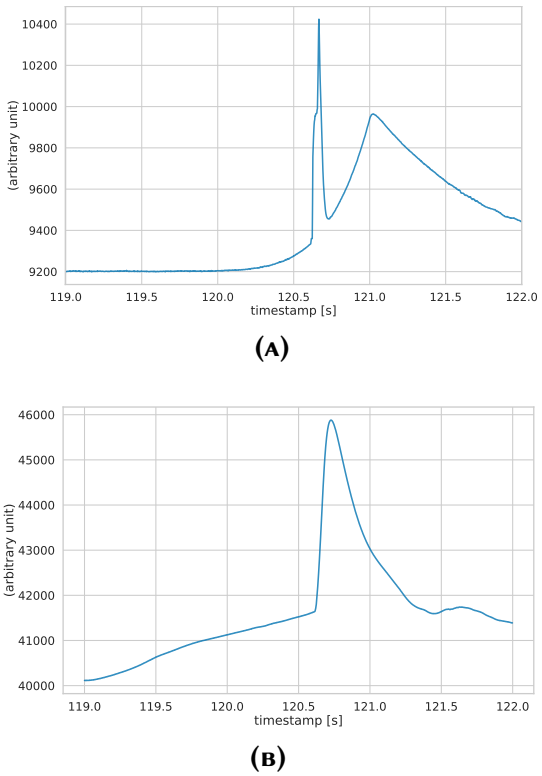


Fig. 3.5: A typical response from physical sensors. Both plots show the measurements for the first sample of the analyte PETN. On the top panel, the sensor response of the UV light sensor can be seen; the bottom panel shows the response of the pressure sensor.

In some cases, a gas sensor might not show a valid response, due to the type of gas being emitted or to some kind of malfunction. This can be determined by e. g. plotting the signal. The response is often times easy to distinguish from a clear response and can be spotted by eye. Figure 3.6 gives an example of sensor UST-5333 (S4) showing no response. The plot is taken from the first run of the analyte Geosit. All data points scatter around the mean value of the signal, which suggests that no valid sensor response is produced. Especially when extracting features from TS data, it isimportant to identify measurements with no valid sensor response for reasons covered in section 3.3.

3.3 Previous methodology

So far, the procedure to classify an analyte according to its explosion hazard was done by extracting features of points in time for each sensor response. A classifier was developed by Konstantynovski *et al.* (n. d.) based on those extracted features, achieving high detection rates. This section will mainly focus on the extraction part of said features. The later course of the thesis will provide an alternative way of feature engineering (introduced in section 2), extending the current work of binary classifiers, applied to a different use case.

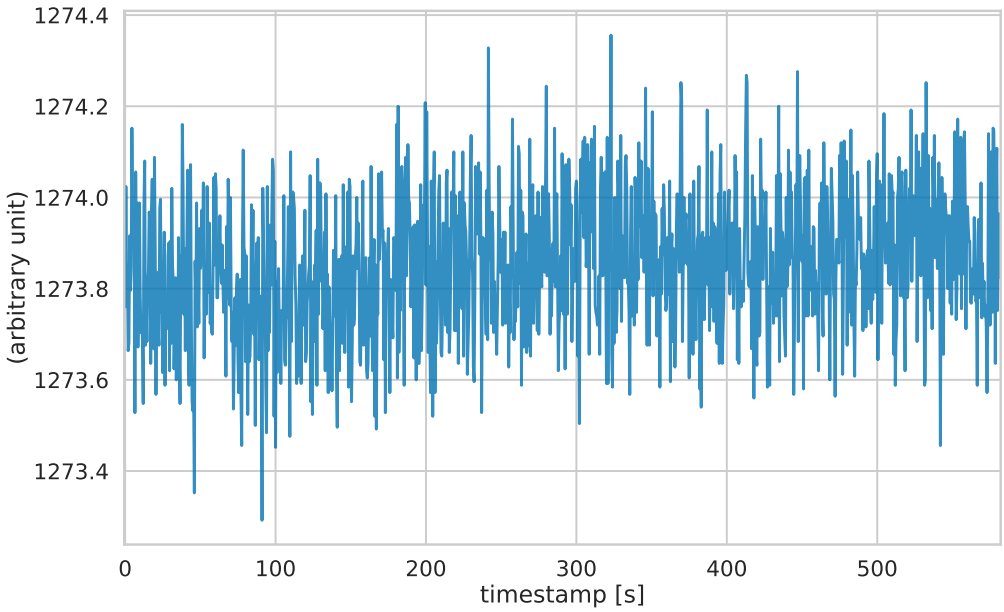


Fig. 3.6: Sensor showing no response to the sample. The measurement originates from sensor UST-5333 (S4) for the first run of the analyte Geosit. Looking at the y-axis it can be seen that all data points scatter around the mean value of the signal, indicating no valid pattern.

The part of extracting features relied mostly on domain knowledge, such as the functionality of the hardware and sensors or certainty about the quality of different sensor responses. Initially, it is required to do an inspection of the data at first and decide on a proper extraction process. The inspection of the TS should correctly identify sensors showing no valid response, as exemplified in fig. 3.6. Many possibilities exist to detect non-valid signals in MOX gas sensors. In the course of the thesis, a method has been developed, showing promising results when comparing the very beginning of the signal response (step 1) with the phase of thermal activation and sampling (step 6–9). For both sections, the Signal-to-Noise Ratio (SNR) can be determined, which is defined by μ/σ. In a second step, the values can be used for the calculation of a threshold with

$$\frac{\mathrm{SNR}_{[0,10]} - \mathrm{SNR}_{[119,522]}}{\mathrm{SNR}_{[0,10]}}$$

whereby a signal with a result of $\geq 0{,}5$ is declared as valid. Figure B.1 on page 65 shows an example for the validation of the sensor response for two different analytes. The first 10 seconds (Step 1) of the signal are chosen as a reference, since the chamber of the gas sensors is still empty. The sensor output of this interval is regarded as noise and can be compared to steps 6–9 to detect any significant difference in terms of SNR. If by this procedure a sensor output was declared as not valid, every feature extracted of this specific TS signal got set to the numeric value of „0“ afterwards.

Up to five features were defined for the extraction process. Depending on the sensor type, only a subset of two features were extracted. For any TS produced by gas sensors, the complete feature set was considered. Physical sensors relied on only the first two in total. The features defined by Konstantynovski *et al.* (n. d.) are:

a) global extremum of the 1st derivative
b) time in seconds for reaching *(a)*
c) value at 121 seconds corrected by the baseline value at 71 seconds
d) value at 149 seconds corrected by the baseline value at 71 seconds
e) time taken to recover to 50 % of the value at 162 seconds

Given the number of 3 physical sensors and 6 gas sensors, 36 features in total could be extracted for each run of an analyte, since each run is multivariate and consists of 9 TS measurements. This process is visualized in fig. 3.7. The plots show an example for both sensor types of the analyte TNEB. It can be seen that a prior inspection of the signal is necessary; using the features extracted from non-valid signals would lead to a distortion in the feature set. Table 3.5 shows an example of the feature set with arbitrary values. For each run of every analyte, all 36 extracted features are being concatenated into a single vector, enabling the use of conventional ML methods.

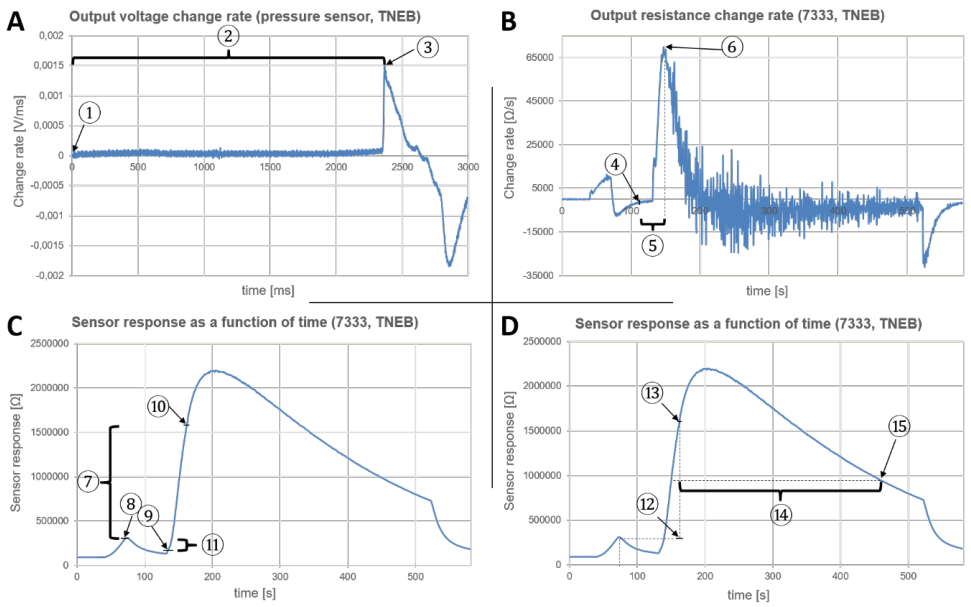


Fig. 3.7: Example for the feature extraction procedure.‡ The two plots in the top demonstrate the extraction of the first two features *(a)* and *(b)* for both sensor types. For feature *(c)* and *(d)*, the plot in the bottom left shows the required baseline value *(8)* and the two different points in time *(9)* and *(10)*. The last plot gives an overview for the extraction of the remaining feature *(e)* with the baseline-corrected point in time *(13)* and the duration *(14)* to recover to 50 % of its value *(15)*.
‡ Figure taken from Konstantynovski *et al.* (n. d.).

Tab. 3.5: Example of the feature set concatenated into single vectors. The table shows arbitrary values for the features as concatenated vectors for the runs of two analytes X and Y. For each run n every feature j of a sensor i is flattened into a single vector. This results in a feature vector with 36 variables of all 9 sensors of every sample of an analyte. In addition, the labels are given for a binary classification task.

		Sensors							
		$S_{i=1}$				S_i			
Analyte	Run	$F_{j=1,i=1}$	$F_{j=2,i=1}$	...	$F_{j,i=1}$	$F_{j=1,i}$	...	$F_{j,i}$	is explosive
X	$X_{n=1}$	*123*	*99*	...	*120*	*30*	...	*50*	1
X	$X_{n=2}$	*40*	*0*	...	*80*	*26*	...	*70*	1
⋮	⋮								⋮
X	X_N	...	...	...	...	...	...	...	1
Y	Y_n	...	...	...	...	...	...	...	0

The obtained features were used to build a binary classifier to distinguish between explosive and harmless substances. The train and test set, as shown in table A.1 on page 64, has been used to create and validate the supervised algorithm. The classifier consists of three successive steps. For each step a complete screening of all features searches for those that are able to exclude the False Positive error type. To find the best possible threshold value for each feature, the Receiver Operating Characteristic curve (ROC) can be analyzed. Figure 3.8 shows an example of feature *(a)* for the pressure sensor. The screening procedure results in a Recall of ~70 %, while still having a False Positive rate of zero. The Recall can be used as a score for this feature to rank against all remaining features in this step. Out of this subset of features with a False Positive rate of zero, the highest scoring, according to its Recall, will get selected. After each step, every sample classified as explosive can be disregarded, since False Positives have been fully excluded. For the remaining samples, the screening procedure will get repeated up to the last third step. Every sample not determined explosive after the last step is declared a benign substance.

The three features selected along with the corresponding threshold values were used to validate the classifier on the test set. Here, a detection rate of 90 % was achieved, with one False Positive misclassified. As a possible improvement for the training set, Cross Validation could be used. This gives the opportunity to evade problematic scenarios like overfitting the training set and would lead to a more robust implementation of the algorithm (Hastie *et al.* 2009, pp. 241 sqq.). To avoid data leakage in this case, a grouped variant of K-fold or Leave-One-Group-Out for the samples of an analyte should be selected (Guts 2018). Further stratification of the folds would take care

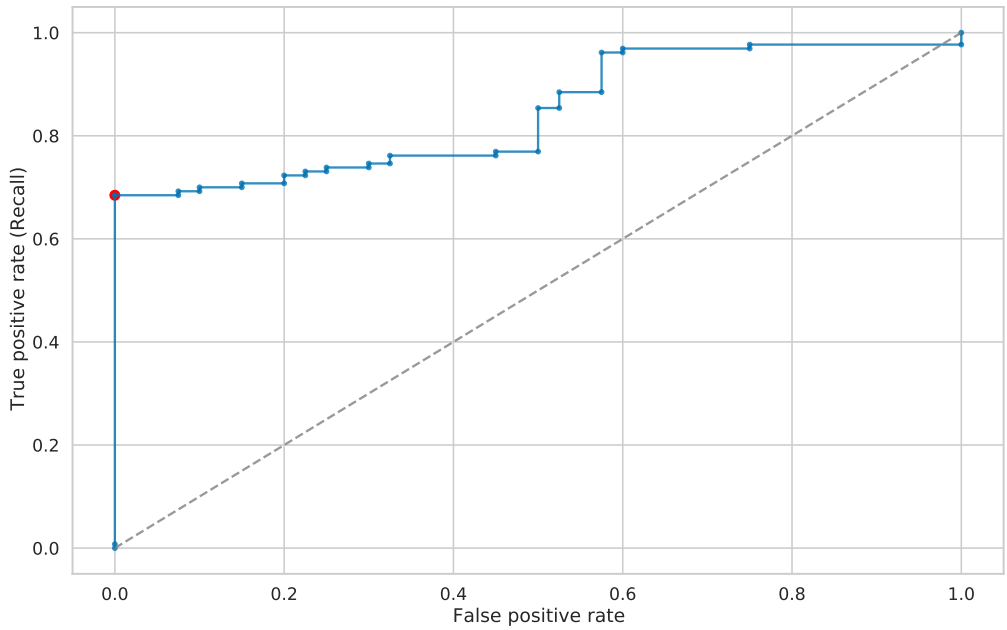


Fig. 3.8: Determining a classifier value with the ROC chart. The example shows all qualifier values and the optimal one (red dot) of feature *(a)* for the pressure sensor. This value is determined by searching for the highest possibly Recall, in this case ~70 %, with a False Positive rate of zero.

of the class imbalance. Alternatively, a 50:50 ratio of positive and negative examples in both sets should be achieved, to avoid the need of stratification or any other re-balancing countermeasures.

4 | Implementation

The implementation of the methods proposed for the dataset (section 2) will be explained in the following sections. The use of ML algorithms can be versatile, and intended use cases are often times very different across the various domains (Martínez-Plumed *et al.* 2019). In general, the planned execution of those algorithms on a dataset is also called a (Machine Learning) *pipeline*, where data is transformed sequential over multiple defined steps. Various guidelines and patterns exist for building those kinds of pipelines. This thesis will make use of one of the most common approaches (Nisbet *et al.* 2018, p. 40): CRoss-Industry Standard Process for Data Mining (CRISP-DM), developed by a consortium of industrial companies, and first introduced by Chapman *et al.* (2000). The process paradigm serves as an independent guidance for robust implementations and applications of ML in an industrial setting. Figure 4.1 shows an illustration of this approach. The model is divided into six successive stages:

i) Business Understanding
ii) Data Understanding
iii) Data Preparation
iv) Modeling
v) Evaluation
vi) Deployment

As indicated by the figure, CRISP-DM shows an iterative pattern. This is in some part due to the nature of ML, being dependent on constant training with more or new data to achieve reliable predictions/results. The arrows visible in fig. 4.1 point out dependencies between particular stages/steps. Given the current progress, some stages are under permanent change and need regular adjustments based on feedback of different stages in the up- or downstream of the process path. Excluded from this is the last and final step „Deployment“, which completes the Data Mining project.

The use of CRISP-DM in this thesis is applicable especially to pre-processing techniques introduced in section 3. Furthermore, the methods described in section 2 can be assigned to their corresponding stage. In case of the first stage of CRISP-DM, „Business Understanding“, already several related publications exist, describing the problem statement in a detailed way (Konstantynovski *et al.* 2017; Konstantynovski *et al.* 2018; Konstantynovski *et al.* n. d.). The proposed solutions and goals can be summarized as a binary classification problem. For that purpose, a specific device was built and improved over time, to be able to produce meaningful data (see section 3.1). This data consists of TS for which particular in section 2.2 several methods have been introduced for the application on the datasets. Regarding „Data Preparation“, section 3.2 has shown

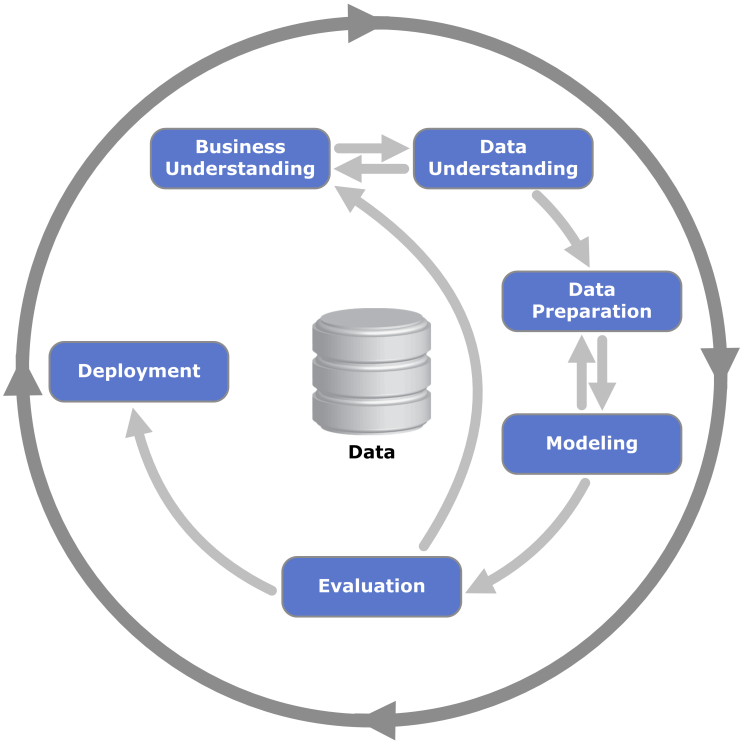


Fig. 4.1: The CRISP-DM diagram[1] as an iterative process. Different stages indicate successive steps to be taken. The process illustrates the approach of a Data Mining project in an iterative manner. It may be necessary to apply knowledge gained through a certain stage in retrospective to past steps, and redefine the outline of those steps again.

steps to transform the data to a more usable format and apply the necessary preprocessing on the signals, e. g. downsampling, and interpolation of outliers. Following this section with stage 3, „Data Preparation“, continues from here on, providing a new method, suitable to work with the ML algorithms selected in this thesis. Afterwards, the BOSS procedure proposed in section 2.2.2 for the extraction of features is shown in context of the dataset. For stage 4, the modeling approaches are shown in their implementation, containing the concepts of LDA and KM introduced in section 2.3. In addition, a small overview is given of methods that were implemented but not capable of reaching the same level of performance. Separately, section 5 will conclude with the evaluation (stage 5) and discussion of the results. Regarding the „Deployment“ at stage 6, the outlook in section 6 can point to possible ways of working in a real-case scenario with the framework developed in this thesis.

[1] https://commons.wikimedia.org/w/index.php?title=File:CRISP-DM_Process_Diagram.png&oldid=506972775, last accessed May 31, 2021.

4.1 Preprocessing

The structure of the available dataset described in section 3 consists of multiple runs for a single analyte. This is due to the reason that in some samples impurities occur, or hardware failures are happening during a run, i. e. sensor malfunction. As a result, the signals of an individual sensor may have a strong deviation across the runs of the corresponding analyte. Several possibilities exist to reduce the influence of erroneous data in the overall result. Section 3.3 presented a method for identifying outliers based on the application of the SNR on specific intervals of the signal.[1] For the application of the methods introduced in section 2 of this thesis, an alternative approach has been taken: given multiple runs of an analyte, an aggregation is applied to reduce the TS down to a single entity; a method called Soft-DTW is used as a geometry measure for a weighted averaging procedure. Soft-DTW was published by Cuturi & Blondel (2017) and is a variant of Dynamic Time Warping (DTW) for the calculation of a distance/similarity measure between TS pairs.

The DTW method tries „[...] to find an optimal alignment between two given (time-dependent) sequences under certain restrictions“ (Müller 2007, p. 69). DTW between two TS x and y can be defined as

$$\mathrm{DTW}(x, y) = \min_{A \in A_{n,m}} \langle A, \Delta(x, y) \rangle$$

where $\Delta(x, y)$ is a cost matrix with the alignment matrix A for both sequences. The objective is to find an optimal warping path in the alignment matrix with the lowest possible cost between both sequences. As a result, every sample point needs to be compared, leading to a computationally complexity of $O(NM)$, with N and M being the total number of points of each sequence (Müller 2007, p. 72). Figure 4.2 shows an example of DTW applied on different signals of the pressure sensor for two runs of the analyte TATP. In the top panel *(A)* both signals are visible, as well as the alignment between them. Panel *(B)* shows the alignment matrix with the optimal warping path found.

In contrast to DTW, the Soft-DTW (S-DTW) variant of Cuturi & Blondel (2017) uses a soft-min operator γ, which acts as a regularization parameter for the degree of smoothing for the path finding process. The S-DTW method is defined as

$$\mathrm{DTW}_\gamma(x, y) = \min{}^\gamma \left\{ \langle A, \Delta(x, y) \rangle, A \in \mathcal{A}_{n,m} \right\}$$

with $\gamma = 0$ being „default“ DTW without smoothing, and a larger γ value corresponding to a stronger degree of smoothing. In case of the data in the thesis, this is of great impact

[1] See also fig. B.1 on page 65 for an exemplary visualization.

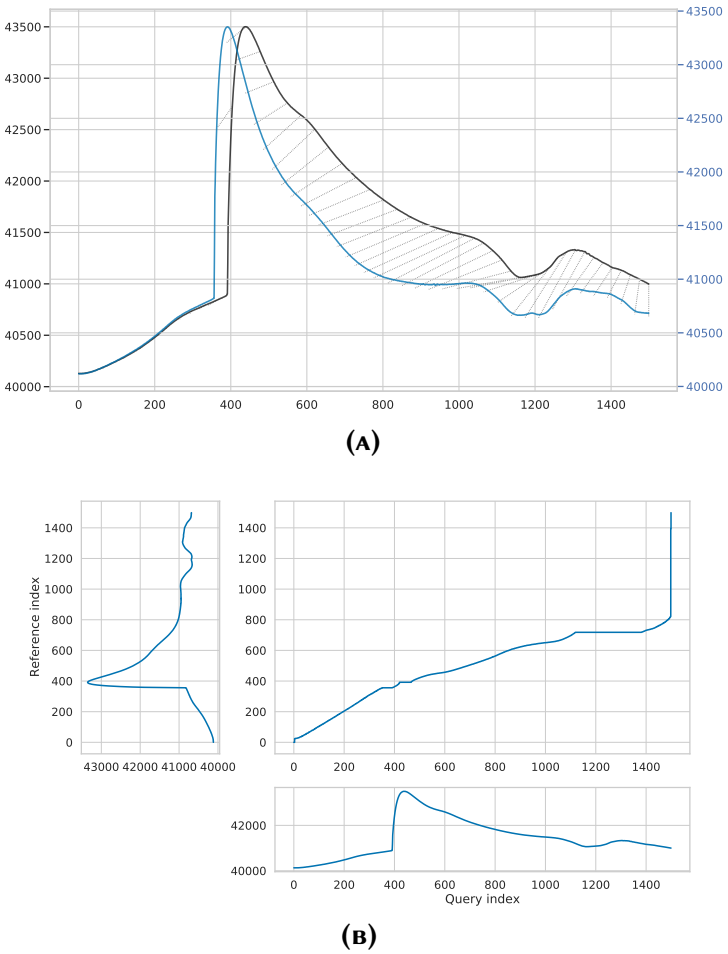


Fig. 4.2: Visualization of DTW applied on two sensor responses.‡ Both signals are being produced by the pressure sensor for different samples of the analyte TATP. In *(A)* the obtained alignment between the signals is visible; *(B)* shows an alignment matrix including the optimal warping path found.
‡ Visualizations created with Giorgino (2009).

for a clean result of the aggregation step. The S-DTW procedure can be implemented as an aggregating clustering mechanism given the measurements for each sensor of an analyte. This aggregation technique is called DTW Barycenter Averaging (DBA) and was introduced by Petitjean *et al.* (2011). The DBA approach is an iterative method, which applies DTW on each sequence pair to create an average representation. In a second step, the overall mean gets updated, based on those representations. Figure 4.3 shows an example for the result of the procedure applied on sensor UST-5333 (S4) for all samples/runs of the analyte HMTD. The first plot *(A)* displays all 10 sensor responses, reduced to the interval of $[119, 522)$ seconds. The self-test procedure $[0, 119)$ and chamber flushing part $[522, 582]$ are disregarded for all gas sensors. This is due to the fact that only the bare reaction of the sample to the thermal activation is crucial for the subsequent feature extraction task. The second plot *(B)* shows the result of the Barycenter Averaging procedure for the runs.

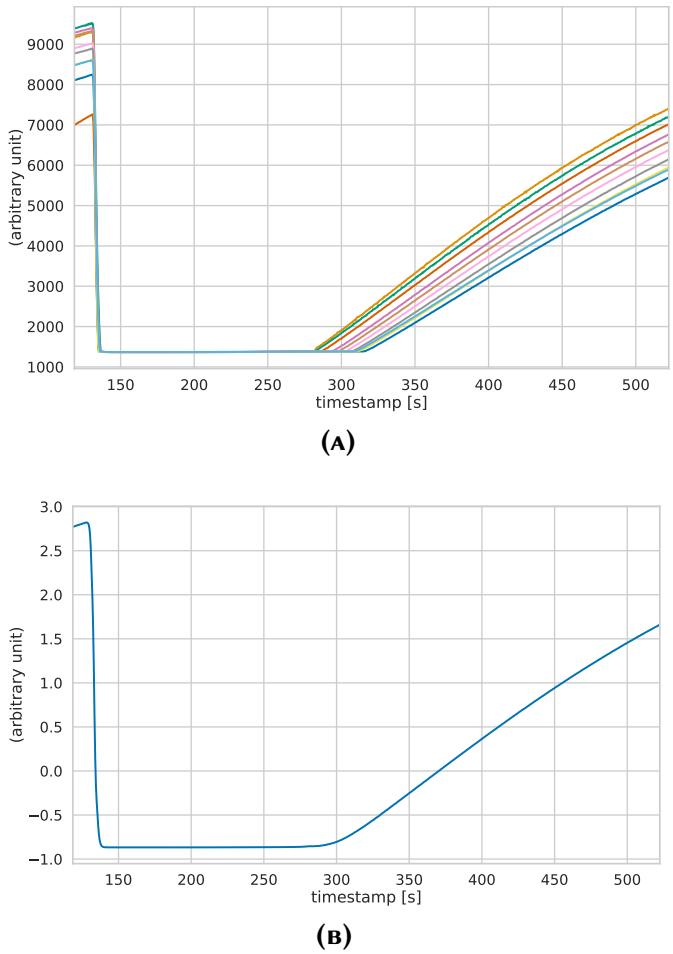


Fig. 4.3: Preprocessing by clustering sensor responses. The example shows all 10 runs of the analyte HMTD for sensor UST-5333 (S4). The interval for gas sensors is reduced to $[119, 522)$ seconds, disregarding the self-test and chamber flushing part of the experiment schedule (table 3.1). In the top panel *(A)* each individual sensor response is shown. The bottom panel *(B)* is displaying the result of the averaging procedure. All TS are scaled to $\mu = 0$ for each case (not shown in top panel) before continuing with the aggregation step.

A key advantage of Barycenter Averaging in conjunction with S-DTW is the flexibility of adjusting the γ hyperparameter.[1] For the dataset in this thesis, a value of $\gamma = 10$ has been chosen after inspecting the results of different qualifier values. Since the TS are equal in length (number of data points) for a specific sensor type (physical/gas), simpler approaches have been considered; for example calculating the arithmetic mean between each data point index across all runs. Compared to Barycenter Averaging, several disadvantages occurred, e. g. the missing smoothing capabilities, or the flexibility of adjusting to (time) shifted signals like with DTW.

The averaging procedure is the last component of the pre-processing pipeline, after the outliers have been removed and the downsampling took place. The application

[1] The Python implementation of Tavenard *et al.* (2020) has been used for the DBA procedure.

of DBA is applied for all runs of an analyte grouped on each sensor. The decision to reduce the amount of multiple measurements/runs down to a single TS for each sensor of an analyte was made, because of the following reasons:

i) Smoothing out leftovers of noise in the sensor responses
ii) Expel any erroneous run by means of aggregation
iii) Get an average representation of each sensor by using the (Barycenter) clustering approach

The transformed and aggregated dataset as a result of this pre-processing pipeline forms the basis for the extraction of features. This corresponds to a portion of the stage „Data Preparation“ of the CRISP-DM model. The next section will conclude this stage with the creation of the feature set.

4.2 Feature extraction

In section 2.2.2 the BOSS algorithm has been introduced. The dictionary-based approach is able to transform each single TS measurement into a BoW representation. Features in the shape of symbols, i. e. strings of characters, are extracted from a rolling subsequence across the full length of the signal. The application of this approach, being a method of the area of ML, can be divided into two elementary steps: *fit* and *transform*. In the fitting procedure the calculation of bins (bin widths) for the coefficients of the DFT takes place; ensuing the creation of symbols depending on the data provided. Usually, for the majority of ML algorithms, the development of the model is done by using a training and a test dataset, for the tuning and validation of performance, respectively. In this thesis, the split of the available data into a train and test set has been adopted from Konstantynovski *et al.* (n. d.).[1] This is due to the reason to achieve comparability and being able to extend the work done so far using the same preconditions. Further, the reasoning for the respective selection of substances in both sets has been adjusted to the binary classification scenario: analytes in the test set are of a very particular (chemical) composition, making predictions more difficult when the model has been fitted with the more common types of substances to be found in the training set.

A transformation of the data, i. e. the feature extraction part, for the training and the test set, takes place after the model has been fitted on the training data. This step is repeated for every individual sensor of the 9 in total. It is necessary to only fit the model once (for each sensor) on the training set, as a common basis to obtain an identical collection of symbols for the test set. Re-fitting the model again on the test set would, in some cases, lead to symbols not present in one set or the other, thus, missing values.

[1] Table A.1 on page 64 lists the complete data, divided into a train and test set.

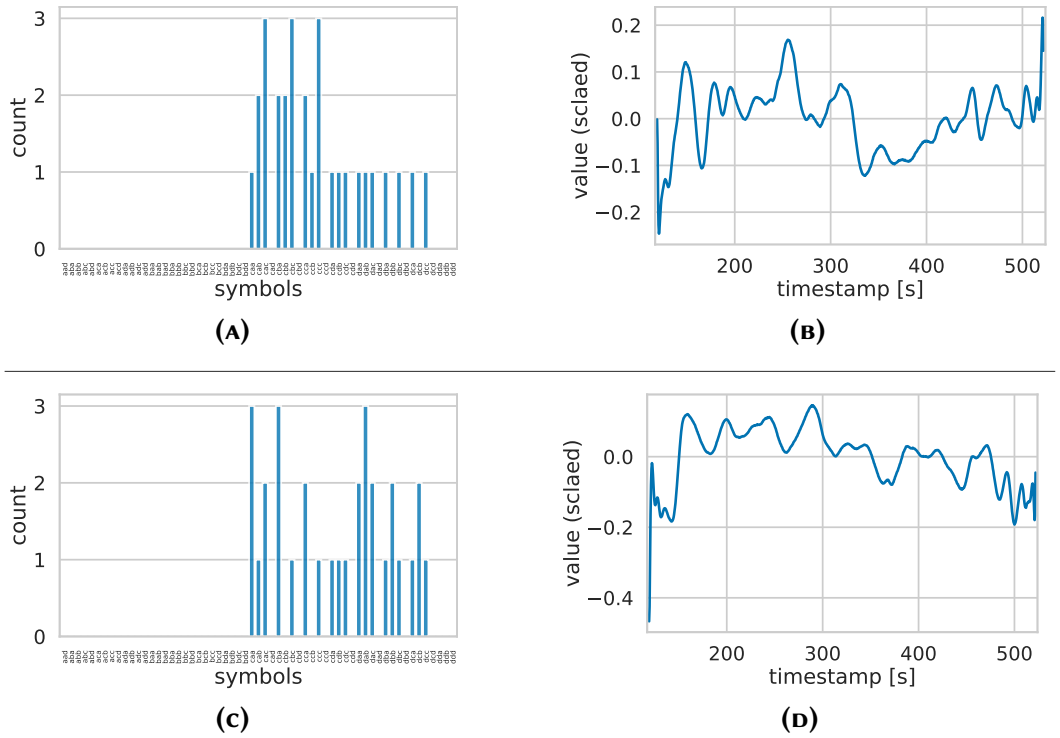


Fig. 4.4: Histograms of the BOSS transformation for Geosit and Urotropine. The Barycenter and histogram for Geosit are visible in the top panel *(A+B)*; Urotropine is shown at the bottom *(C+D)*. Both Barycenters display a highly identical pattern, which also becomes clear when looking at the histograms of the the BOSS transformation.

The fitting and transformation of the Barycenter TS data is done group-wise on each sensor. Figure 4.4 shows an example of the result for the transformation of the analyte Geosit *(top)* and Urotropine *(bottom)* for sensor UST-5333 (S4). Both Barycenters *(B+D)* show a very similar pattern, indicating no valid response of the sensor (as discussed in section 3.2.3). The BOSS model is able to capture this similarity, which can be seen when comparing both histograms *(A+C)*. Another additional example is viewed in fig. 4.5. Here, two Barycenters of TNT *(top)* and Tetryl *(bottom)*, both very much alike, are shown for the same given sensor. The histograms indicate a high similarity between both TS. In contrast, compared to fig. 4.4, the distribution of symbols/features varies between both groups. As can be seen in the examples, the BOSS transformation is able to capture different patterns and similarities in the TS data.

Regarding the customization of the BOSS model, the following hyperparameters were applied for the application on the dataset of this thesis:

- `word_size` = 3
- `window_size` $\in \{0.15, 0.25\}$
- `window_step` = 1
- `n_bins` = 4

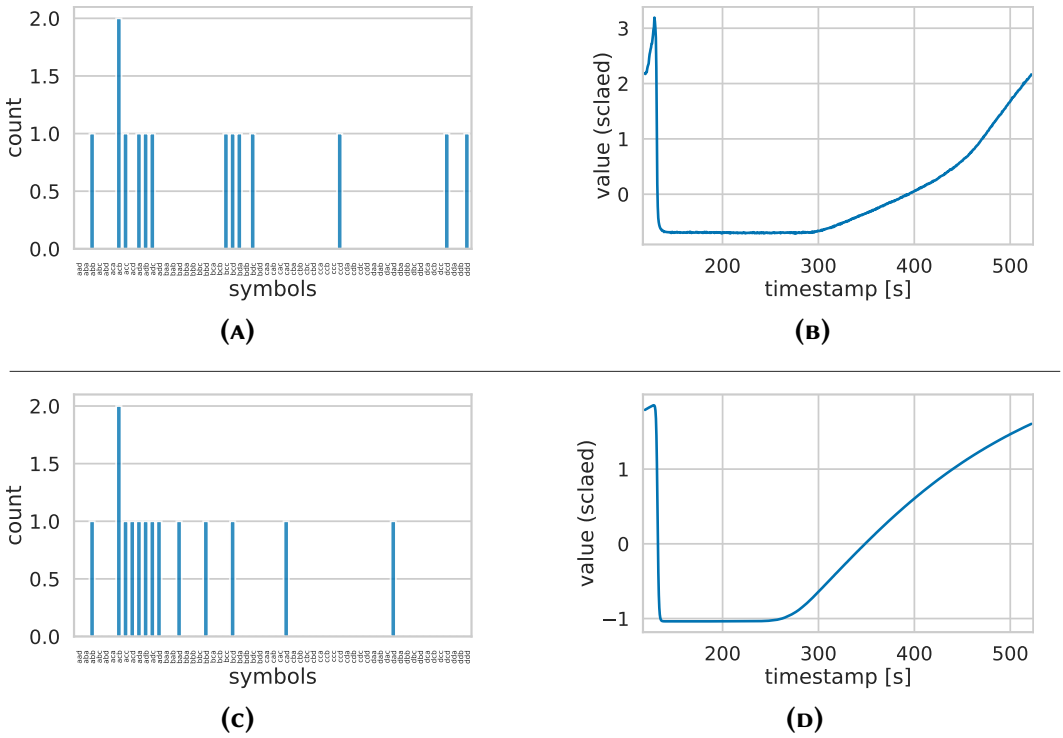


Fig. 4.5: Histograms of the BOSS transformation for TNT and Tetryl. The Barycenter and histogram for TNT are visible in the top panel *(A+B)*; Tetryl is shown at the bottom *(C+D)*. Both Barycenters display a highly identical pattern, which also becomes clear when looking at the histograms of the BOSS transformation.

An overall word size with a value of „3“ (string length) has been used. For physical sensors, each subsequence was created with a window size of „0.15“, which corresponds to a fraction of 15 % of the amount of all sample points in the TS. For gas sensors, the window size has been increased to 25 %. The step size of the sliding window was left at the default value of one sample point. The rationale for the selection of the value for the word size hyperparameter is based on the observation that, most of the time, only a few different patterns of sensor responses existed. A higher word size would lead to a redundant, unnecessary complex (feature) representation, and a lower value is generally preferred for this hyperparameter (Schäfer 2015a, p. 118). By inspecting and comparing results of several hyperparameter values for the BOSS model[1] across all analytes for different sensors, the most suitable value could be found. This step relied partly upon the „Data Understanding“ stage of CRISP-DM, which included an extensive data review process. Another customization of the BOSS model could have been made via the hyperparameter `n_bins`, which corresponds to the depth of available characters $\{A_1, B_2, \ldots, Z_{26}\}$ in the collection for the discretization/binning of the coefficient values. No measurable improvement could be determined when

[1] The Python implementation of Faouzi & Janati (2020) has been used for the creation of the BOSS model.

comparing the results while increasing or decreasing the default value of „4“, implying characters A–D. Regarding the window size hyperparameter for the subsequences, the value was increased for the gas sensors, because of different time scales between both sensor types. The frequency for the downsampled gas sensors was 2 Hz, and for the physical ones 500 Hz (section 3.2.2). Due to the nature of gas sensors, any pattern in the signal, i. e. concentration of various gases in the air, extends over a larger time range, when compared to physical phenomena, e. g. a sudden spike in pressure, or flashes of light. Therefore, the length of the subsequence (window size) should be adjusted according to the circumstances, to be able to cover (visible) patterns in the TS measurement (Schäfer 2015a, p. 118).

The obtained feature set has the same structure like the one described in section 3.3 at table 3.5. For every analyte, all feature vectors from all 9 sensors have to be concatenated into a single vector. Table 4.1 shows an excerpt of the BOSS transformation for the training set. The numeric value for every available symbol (sorted alphabetically)

Tab. 4.1: Excerpt of a feature set generated by the BOSS model. The results of the transformation on the training set is shown. The count value for each extracted symbol, with respect to the corresponding sensor, is contained in a (sparse) feature matrix. An additional column containing labels with class memberships $y \in Y$ is given for a supervised learning scenario or, in this case, to be used as external information in a clustering evaluation.

	Symbols									
	AAA			AAB				DDC		
Analyte	S1	…	S9	S1	…	S9	…	S1	…	S9	Label
AN	8	…	0	3	…	0	…	0	…	0	y
BP	0	…	0	0	…	0	…	0	…	0	y
⋮											⋮
Tetryl	0	…	0	0	…	0	…	0	…	0	y
Tovex	1	…	0	2	…	0	…	0	…	0	y

is listed with its corresponding sensor (S1, S2 ... S9). In total, 64 symbols have been extracted for each individual sensor. After concatenating every vector into a single one for every analyte, the matrix grows in size up to 576 columns (features).

This section concludes the „Data Preparation“ stage of the CRISP-DM model. The created feature set contains discrete numeric values, representing counts of symbols; it shares the same properties as the BoW model. The next section will demonstrate the application of LDA and KM utilizing this data structure to solve specific use cases in the „Modeling“ stage. The methods used for the creation process of a model, as well as the different use cases will be presented in the next section.

4.3 Application scenarios and selection of models

The provided features in the train and test set, containing the concatenated vectors for each analyte, will be used for the creation of the LDA model. This first step in the ML pipeline, after the preprocessing took place and the BOSS features have been generated, serves as a technique for reducing the dimensionality of the (sparse) BoW representation, as well as extracting new features from the data, i. e. latent topics. The reduction of the number of features is a necessity when working with a clustering algorithm like KM (*Curse of Dimensionality*, section 2.2.1), which will be applied subsequently to the LDA method. The new representation obtained through the Topic Modeling approach will be used in a clustering step with KM, utilizing different class labels depending on the selected use case; therefore, the use cases will be explained accordingly.

Figure 4.6 shows a schematic view of the data and ML pipeline deployed for the work in this thesis. At the point of reaching the stage of fitting both models, LDA and KM, the process is split up for the specific use cases.

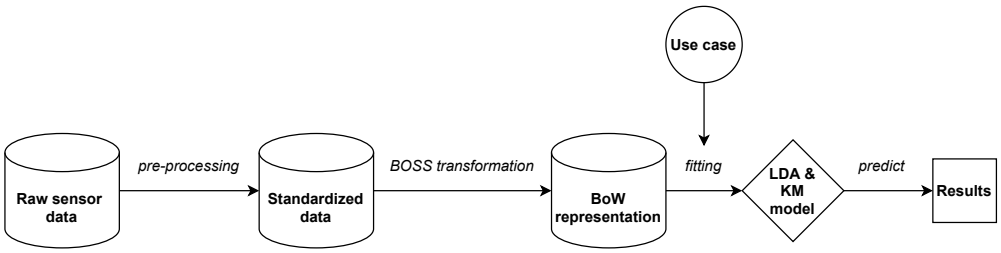


Fig. 4.6: Schematic view of the deployed data and ML pipeline. The pipeline shows the process of preparing the raw sensor response data for the transforming step with the BOSS method. Afterwards, depending on the given use case, both models, LDA and KM, are fitted and evaluated by comparing the performance of using clustering metrics in conjunction with the (external) class labels.

4.3.1 Use cases

In total, two use cases have been defined, based on the application of domain knowledge. Both are being implemented independently, and only share the feature set of the BOSS transformation as a common ground. The creation process comprises a pipeline, which performs the training and evaluation part of the model in dependence of the supplied class labels. The work done so far in the publications related to this dataset, focused on the binary classification setting of explosive and non-explosive substances. Section 3.3 presented the 3-step classifier algorithm developed by Konstantynovski *et al.* (n. d.) regarding this task. Further, some approaches have been made to identify individual substances or groups of substances in terms of their chemical structure and similarity among each other. Konstantynovski *et al.* (n. d.) tested Principal Component Analysis

as a dimension reduction technique to obtain a 3-dimensional representation (first three components) to visualize and search for any significant patterns, i. e. groups/clusters. This thesis will continue with the implementation of methods for the identification task of chemical substances.

In addition to the objective of a binary classification scenario, two possible use cases arise, through the expertise of domain knowledge, for the chemical substances in the dataset:

A) Identification of high-energetic explosives
B) Assigning substances into groups of similar chemical structure

For both cases, only the train set has been used. This is due to the reason that the identification and clustering of substances contained in the test set is less meaningful, given the rather special chemical compositions. Nevertheless, section 5.1.2 added an evaluation for the new proposed methods to test their usefulness as a binary classifier for the distinction between explosive and benign analytes.

The first case *(A)* of identifying high-enegetic substances can be understood as a gradation of the explosive potential. Figure 4.7 shows a taxonomy of the explosives, in which high-energetic[1] substances are listed as „primary explosives“, while lesser energetic substances are known as „blasting explosives“ or „propellant charges“. The following groups of analytes can be assigned for this use case:

i) very high-energetic: HMTD & TATP
ii) high-energetic: PETN
iii) explosive: AN, BP, RDX, Semtex, Tetryl & TNT
iv) insensitive explosives: Geosit, Pikramid, TNEB & Tovex

The assignment of the groups can be adapted to numeric labels as shown in table 4.1, in order to evaluate the performance of ML models with the specific clustering performance metrics of section 2.3.3. As for the second use case *(B)*, the substances can be sorted into the following 7 (chemical) groups:

a) Anorganic salt: AN, Geosit & Tovex
b) Nitro aromatic: Pikramid, TNT & TNEB
c) Nitramine: RDX & Tetryl
d) Nitrate ester: PETN
e) Peroxide: HMTD & TATP
f) Plastic explosive: Semtex
g) Sulfur, charcoal and potassium nitrate: BP

[1] The difference between high and very high energetics is rather neglectable but listed for the sake of completeness.

Given the different class labels, two pairs of the combination of LDA and KM will be fitted separately to the dataset. Once trained, each pair is able to predict new, unseen data, adapted to a specific use case.

The next section will present the specific implementation of both methods, LDA and KM, to build a unified ML model for the classification of an analyte. The evaluation of the results will be done in section 5.

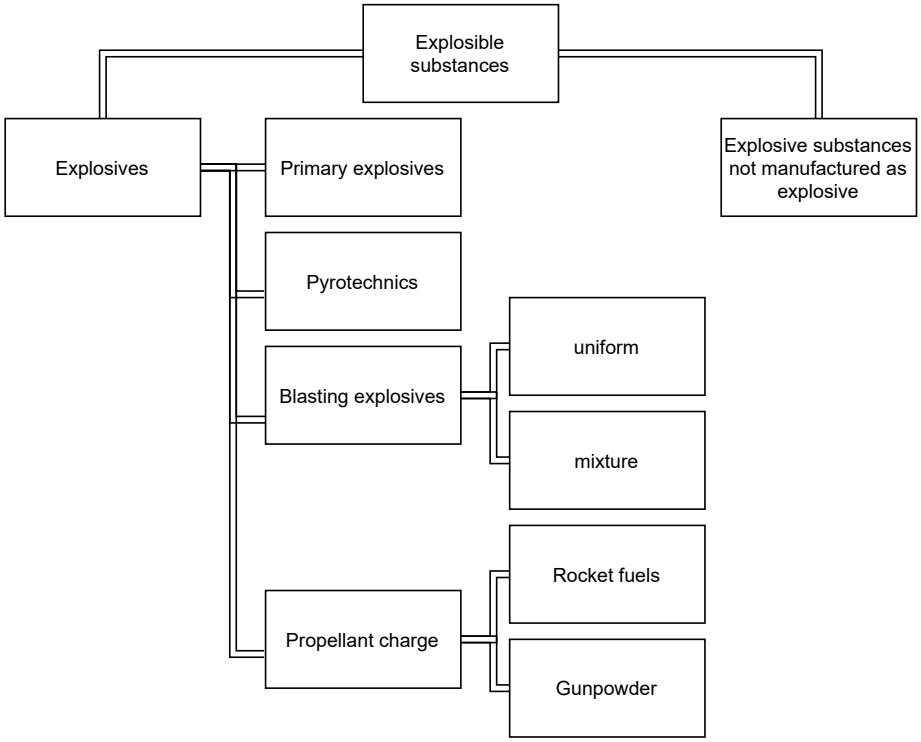


Fig. 4.7: Taxonomy of explosives by type of use.‡ As an example, high-energetic substances are listed as „primary explosives“. There are no substances from the categories „pyrotechnics“ and „rocket fuels“ in the existing dataset for this thesis.
‡ Own figure based on Konstantynovski (2018, p. 17).

4.3.2 Selected methods

This section will outline the reasons of selecting the ML methods LDA and KM.[1] The BoW based sensor data, transformed by the BOSS model, has been used for the application of LDA. The clustering procedure KM was used for the classification of analytes in a subsequent step, based on the obtained topics of the LDA method. A Grid Search was utilized to find the optimal hyperparameter values for both algorithms, and to fit the models according to the given use case.

[1] The Python implementation of Pedregosa *et al.* (2011) has been used for the creation of both models.

The reason for choosing LDA as one of the essential methods in the work of this thesis was largely dependent on the BoW model obtained through the BOSS transformation. As described in section 2, the representation of documents in the form of a BoW model is one of the most common approaches in the area of Natural Language Processing and Text Mining. The LDA method provides a way of reducing the dimension of this structure, while extracting hidden, unknown topics, providing semantic relevance. In contrast to the TS data in this thesis, which is being transformed via the BOSS method into a BoW representation, there is no semantic meaning present, out of any topic extracted. However, the LDA method can still be useful, without the need of semantic relevance, and especially to reduce the dimensionality of the feature set in an efficient way. Alternative ways of working with this kind of discrete data structure exist, as for example Singular Value Decomposition or Non-Negative Matrix Factorization. Although compared to LDA, both methods do not grant the same degree of freedom in regards to the hyperparameter combinations, when choosing or searching for the best fitted model. The advantage of LDA in this case are two priors η and α, allowing for a higher granularity and influence on the ML model during the creation phase. For the application of LDA in this thesis, the following hyperparameters have been defined for the search space:

- $K \coloneqq$ `n_components` $\in \{2, 3, \ldots, 32\}$
- $\alpha \coloneqq$ `doc_topic_prior` $\in \left\{K^{-1}, 1, 2, \ldots, 24\right\}$
- $\eta \coloneqq$ `topic_word_prior` $\in \left\{K^{-1}, 1, 2, \ldots, 24\right\}$

Both additional hyperparameters α and η can (optionally) be set for the implementation of LDA. As described, the data in this thesis does not provide any semantic indications regarding the extracted topics, and therefore no prior knowledge for both, the topic-word or document-topic distribution. For this reason, a hyperparameter search for both priors has been included, in addition to the number of topics. In the default case for both priors, a normalized value $1/K$ is used, depending on the number K of topics.

The decision of choosing an additional unsupervised learning algorithm as a predictor for the class membership was mainly due to the reason of flexibility regarding the number of resulting classes, i. e. clusters. Using a clustering method provides the ability to freely define a broad hyperparameter range for the number of groups/classes the ML model can create in a given use case. This allows for a more versatile model, as opposed to setting a fixed number of topics or clusters equal to the number of classes. In terms of the implementation for KM, the hyperparameter K for the number of clusters was not set to a fixed value for the search procedure, but a predefined search space, similar to the implementation of the LDA model. This approach does not limit the amount of possible clusters. Although regarding the number of clusters produced by KM, the

minimum value would at least be at two clusters, and the maximum limited by the number of observations $N - 1$.

Using this kind of implementation, the result may obtain more clusters than possible classes. In the case of the data in this thesis, the use cases defined groups of analytes based on domain knowledge. Although given the latent structure of the feature representation and the nature of chemical substances, allowing more clusters than classes might lead to a more robust result, since there still might exist some subgroups that are not covered by the class labels. As long as the resulting clusters are homogeneous and the clustering partition still provides an adequate completeness, this approach can lead to a better performance. By using the unsupervised approach, instead of a supervised one, a more flexible implementation is reached.

This section concludes the „Modeling“ stage of CRISP-DM. Section 5 will present the evaluation of the results obtained in the Grid Search procedure. Several alternatives to Grid Search exist, for example Random Search or more sophisticated approaches, like Bayesian Optimization, which might have some performance advantages regarding efficiency and computational load. For the hyperparameter search in this thesis however, Grid Search, as a comparatively simpler method, was chosen, since the defined hyperparameter space is rather shallow and less complex. Moreover, an advantage of Grid Search is the evaluation of all possible hyperparameter combinations and therefore guaranteed finding of the best possible qualifier values for the hyperparameters provided in the search space.

5 | Results and evaluation

This section represents the „Evaluation“ stage of CRISP-DM and provides the results for different classification scenarios, as well as a comparison between the different models created. At first, the performance of the selected methods will be shown in context of the respective use case. Afterwards, a discussion of the results concludes this section.

5.1 Performance of selected methods

5.1.1 Identification of substances

In contrast to the detection of explosives, the identification of chemical substances is implemented after the classification between explosive and benign analytes took place. Since an already reliable solution exist, the models created for both use cases of identifying analytes do not include any non-explosive materials; assuming that these benign substances have already been sorted out in the process chain. This leads to an increase in performance and simplification of the model creation process, since fewer classes have to be considered.

The V-Measure has been taken as a first indication for the goodness of the clustering partition. Being the harmonic mean between homogeneity and completeness, the V-Measure is more easier to interpret, compared to the other available scores. In addition, the model with the lowest number of K clusters has been chosen, in the case of a tie, avoiding unnecessary complexity.

***(A)* High-energetic explosives.** The assignment of four possible class labels in this use case was listed in section 4.3.1. All 13 explosives in the training set have been used for this objective. Table 5.1 shows the result of the hyperparameter search. At the top, the best scoring entry shows a slightly better score than the following entries. In particular, the homogeneity score („h.“) is able to reach the maximum value. In return, the completeness metric („c.“) is worse, compared to the other two. It has been decided to choose the first entry for the hyperparameter combination. Although the other two entries show better results for the remaining metrics, a higher homogeneity score is of greater impact. This results in the following six clusters:

i) Geosit, Pikramid, TNEB & Tovex
ii) AN, RDX & TNT
iii) HMTD & TATP
iv) PETN
v) Semtex
vi) BP & Tetryl

Tab. 5.1: Grid Search for the case of identifying high-energetic explosives. An excerpt of the top three and the last result shows the performance by ranking the entries according to their V-Measure score.

	Hyperparameters									
	K-Means	LDA			Scores					
Nr.	K	K	α	η	Rand	Mut. Inf.	V-M.	h.	c.	F.-M.
1	6	5	4	0,20	0,59	0,72	0,84	1,00	0,72	0,71
2	3	5	3	0,20	0,67	0,78	0,83	0,71	1,00	0,80
3	3	7	2	0,14	0,67	0,78	0,83	0,71	1,00	0,80
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
213.125	2	9	18	5,00	−0,09	−0,15	0,05	0,04	0,07	0,33

As opposed to the predefined number of four classes, two additional groups emerged. First of to notice, is that insensitive explosives in cluster *(i)*, the high-energetic explosive *(iv)*, and both very high-energetics *(iii)* still remain in their own clusters. In contrast, the group of the remaining explosives is split up accross cluster *(ii)*, cluster *(v)*, and cluster *(vi)*. At least the membership of AN in cluster *(ii)* can not be recognized in the context of the use case, since this kind of substance should not be rated on the same level of energetic potential like RDX and/or TNT, but below, isolated in a single cluster. Regarding the remaining last two clusters *(v)* and *(vi)*, all three analytes should be combined into one cluster, which would increase the overall completeness score of the clustering partition. Although, as an important finding, all insensitive explosives (lowest degree of energetic potential) and high-energetic substances could be clearly separated from the rest.

In addition, it has been tested if a higher granularity can be applied to this use case. For the class of the remaining explosives, a division into 3 subgroups can be made, resulting in a total of 6 classes. Those new groups of classes are: *(1)* TNT & RDX; *(2)* BP, Semtex & Tetryl; and *(3)* AN. The first two represent the highest degree of energetic potential within this selection, beneath the high-energetic explosives, and the last one can be placed above the insensitive explosives. Table 5.2 shows the result of the Grid search procedure for the case of the expanded grouping, ranked after the V-Measure („V-M.“) score. The first entry indicates a good performance score, compared to the other 2 alternatives. The hyperparameters of the first entry have been adapted for the model creation of the expanded variant of this use case. As a result, the following clustering partition has been obtained:

i) Geosit, Pikramid, TNEB & Tovex
ii) RDX & TNT

iii) HMTD & TATP
iv) PETN
v) Semtex
vi) BP & Tetryl
vii) AN

As can be seen, it is possible to isolate AN to its own cluster. There are no more differences between this partition and the other one from before; Semtex *(v)* is still separated from the actual group *(vi)* originally assigned by the class labels.

Tab. 5.2: Search results for the expanded use case of high-energetics. The result table has been sorted according to the V-Measure metric.

	Hyperparameters									
	K-Means	LDA			Scores					
Nr.	K	K	α	η	Rand	Mut. Inf.	V-M.	h.	c.	F.-M.
1	7	5	4	0,20	0,89	0,89	0,96	1,00	0,92	0,90
2	7	6	4	0,17	0,89	0,89	0,96	1,00	0,92	0,90
3	8	5	4	0,20	0,67	0,75	0,91	1,00	0,84	0,74
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
213.125	2	9	18	5,00	−0,09	−0,20	0,07	0,05	0,14	0,19

The topics created by LDA can be visualized using a specific procedure called LDAvis, developed by Sievert & Shirley (2014). Using this approach it is possible to see the closeness and similarity between all topics and the most relevant terms, i. e. symbols, in each individual topic. In this thesis, the interpretation of extracted symbols is more difficult compared to words/terms of natural language, due to lack of semantic meaning. However, there may be multiple symbols of a specific sensor which occur very frequently. Therefore, at least very predominant sensors can be interpreted as a sign of importance for a given topic.

For the topic comparison and interpretation of symbols, the topic with the highest probability has been assigned to each analyte. Since LDA produces a distribution of topic memberships, this intermediate step is helpful for comparing topics. Figure 5.1 shows the visualization of LDAvis[1] for the high-energetic use case. It is apparent that each topic is well separated and there is no overlap between topics, which means that each topic contains its own distinct set of symbols. Using the described hard-cluster approach by assigning each analyte to the topic with the highest probability, the visualization can be used as a first indicator for the goodness of the clustering partition. Only topic 3, containing all four insensitive explosives, shows a clear result in this case. The rest of

[1] The Python implementation at https://pypi.org/project/pyLDAvis/ has been used for LDAvis.

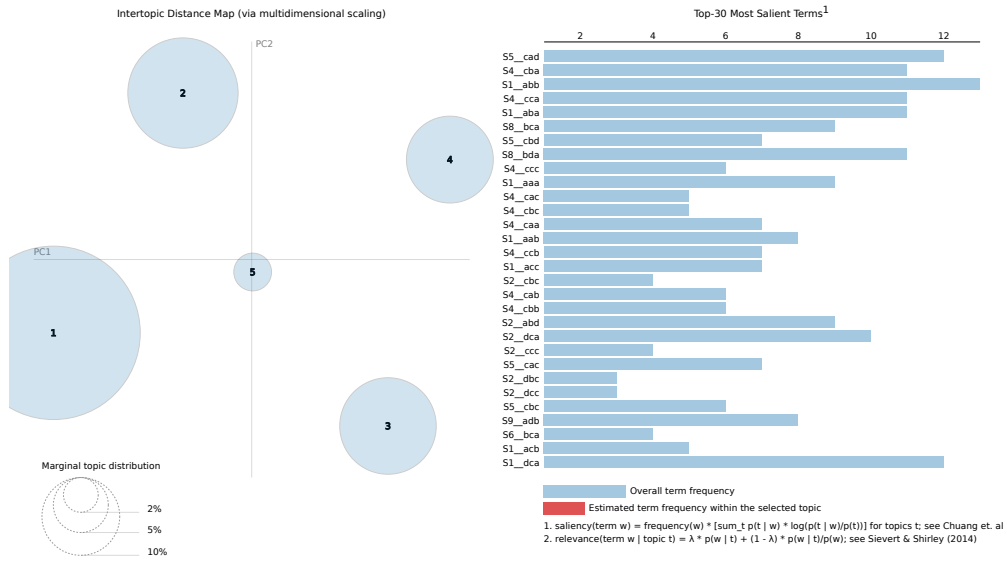


Fig. 5.1: Result of LDAvis for high-energetic materials. All topics are well separated from each other.

the topics are mixed up between the high-energetic and remaining explosives, thus, can only be properly separated by the subsequent KM algorithm. One factor worth emphasizing is that the top-10 most relevant symbols in topic 3 originate solely from gas sensors. In contrast, the top-10 for topic 2 has only one symbol of a gas sensor present, but nine of physical sensors. In addition, no analyte was assigned to topic 1 using this hard-clustering approach.

***(B)* Similar chemical structure.** For the use case of identifying substances of similar chemical structure, section 4.3.1 defined seven groups for the explosives in the train dataset. Table 5.3 shows the result of the Grid Search procedure for the hyperparameters. Values of different rankings have been tested for the implementation. The (adjusted) Mutual Information („Mut. Inf.“) score has been chosen as the criteria in this case, selecting the hyperparameters of the first entry for the implemented model. Different hyperparameters ranked by the remaining scores could not lead to satisfying results. Overall, it was not possible to achieve a clustering partition with maximum homogeneity, while still providing a low number of clusters. For this use case, the following clustering partition has been received:

i) TNT
ii) Pikramid & TNEB
iii) Geosit & Tovex
iv) TATP
v) AN

vi) BP
vii) PETN, RDX & Tetryl
viii) HMTD
ix) Semtex

It can be seen that cluster *(vii)* is impure. In addition, the partition in its entirety features many clusters with only one member, resulting in a high homogeneity score („h."), and a lower completeness („c."). Besides Semtex (*ix*) and BP (*vi*), all other groups are incomplete and spread across the clustering partition. This use case is the most difficult one for the modeling approach, since the definitions of similar chemical structures are not as unambiguous when compared to the the first use case, in which substances were ranked after their (measurable) energetic potential.

Tab. 5.3: Results of the grid search for the use case of chemical structures. The performance of the results has been determined by the (adjusted) Mutual Information score.

	Hyperparameters									
	K-Means	LDA			Scores					
Nr.	K	K	α	η	Rand	Mut. Inf.	V-M.	h.	c.	F.-M.
1	9	18	5,00	0,06	0,42	0,49	0,86	0,92	0,81	0,47
2	8	16	8,00	0,06	0,42	0,46	0,84	0,88	0,80	0,47
3	8	18	7,00	0,06	0,42	0,46	0,84	0,88	0,80	0,47
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
213.125	3	28	0,04	2,00	−0,14	−0,25	0,22	0,17	0,34	0,06

The quality of the results of LDAvis differ compared to the use case from before. Figure 5.2 shows the visualization of the topics obtained for the second use case. Regarding the topic memberships, BP is the only member of topic 2, and well separated from the rest, identical to the given class labels of the use case. In addition, the top-10 symbols in this topic originate from gas sensors only. Both Anorganic Salts, Geosit and Tovex, are the only members of topic 3, although AN is missing as the third and last member. Another topic worth mentioning is topic 16 containing the two Nitro Aromatics Pikramid and TNEB, but without the third member TNT. Compared to the predictions with KM, it can be seen that those two missing analytes, AN and TNT, could not be correctly assigned in the subsequent clustering step either. Topic 10 in particular, is the biggest topic with up to six members. The two analytes, HMTD and TATP, are both assigned to separate, single topics.

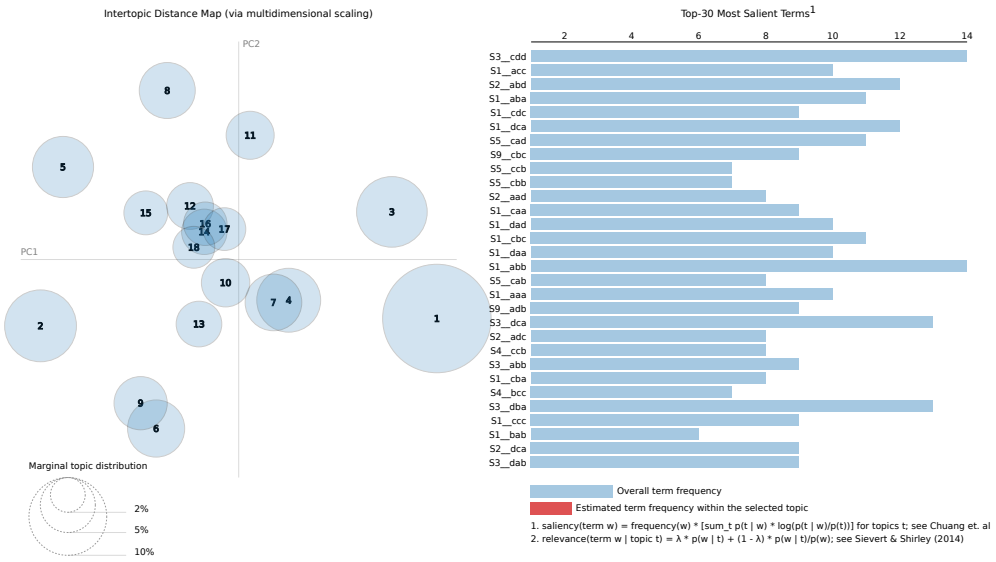


Fig. 5.2: Topics for the second use case visualized by LDAvis. Some overlapping is visible, while some of the other topics are well separated.

5.1.2 Binary classification task

In this thesis, the work was focused on identifying substances based on the two given use cases. As an addition, the developed method was tested in its capability to distinguish between explosive and benign analytes. This was done to allow a performance comparison with the binary classifier of Konstantynovski *et al.* (n. d.). Table 5.4 shows an excerpt of the best three and worst scoring hyperparameter combinations in the Grid Search procedure on the training set. First, the results were sorted by

Tab. 5.4: Hyperparameter search results for the binary classification task. For the ranking of the models, V-Measure has been chosen as a first indicator. The scoring in this case is uniform across all scores, therefore the lowest value for both, the hyperparameter $K^{(\text{clusters})}$ and $K^{(\text{topics})}$, was taken as an additional, second performance indicator.

	Hyperparameters									
	K-Means	LDA			Scores					
Nr.	K	K	α	η	Rand	Mut. Inf.	V-M.	h.	c.	F.-M.
1	3	5	0,20	19,00	0,50	0,67	0,7	1	0,54	0,76
2	3	5	0,20	20,00	0,50	0,67	0,7	1	0,54	0,76
3	3	5	0,20	21,00	0,50	0,67	0,7	1	0,54	0,76
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
290.625	2	25	0,04	0,04	−0,03	−0,07	0,0	0	0,00	0,61

the V-Measure („V-M.“), which has been predominantly used as an indicator for the trade-off between a clustering partition showing high homogeneity („h.“), while still having a good overall completeness („c.“). All metrics clearly show a very unified scoring across the top rated models. As an additional criterion, the list was sorted in ascending order by the number of cluster $K^{(\text{clusters})}$ produced by the KM model, and the number of topics $K^{(\text{topics})}$ of the LDA model, to filter out unnecessary complex models (high number of clusters and/or topics), while still reaching the same level of performance. In this case, the values of the 1st entry shown in table 5.4 were used, resulting in the following clustering partition:

i) AN, BP, HMTD, PETN, RDX, Semtex, TATP, TNT & Tetryl
ii) Geosit, Pikramid, TNEB & Tovex
iii) Abrasive Cleaner, Sugar, Urea & Urotropine

As can be seen, no explosives and benign analytes are members of the same cluster, leading to the highest possible homogeneity score of „1“. This has significant importance for this use case, since a False Negative would have great consequences in the decision-making process for the risk potential of an unknown substance. Another fact to note is that while only two classes have been assigned (binary scenario), at least three clusters led to the best performing model(s). Here, a similarity to use case *(A)* is visible, in which insensitive explosives were defined as an isolated group in advance, found in cluster *(ii)*. All benign analytes remain together in their own, separate cluster *(iii)*, and the remaining explosives in cluster *(i)*. The class labels assigned for this use case consisted only of the two numeric values „1“ and „0“ for explosive and benign substances, respectively. The pattern found here, of multiple (hidden) subgroups, was described in section 4.3.2 as an advantage of the unsupervised approach, i. e. a variable number of resulting classes.

For the binary classification task, the test set was used as an additional validation step. The performance compared to the 3-step classifier algorithm was much worse, and the capability of the model to distinguish explosive from benign substances can not be recognized. Although it could be proven that the approach can lead to useful results and further improvements, e. g. increase in size of the training samples, might help.

5.2 Discussion

The evaluation showed very promising results regarding the first use case of identifying high-energetic substances. This can be beneficial in a real-life scenario, where those substances with a high-energetic potential pose a particular danger. The second use case was more difficult to implement, since when compared with the first use case or

the detection of explosive materials in general, the provided definitions do not grant the same degree of discriminatory power. Especially for some analytes, multiple (cluster) memberships may be possible, since details about the chemical structure of a substance are often times not as deterministic. The definitions of the chemical groups presented here are of the most general case. Variations are possible, but the performance shown is a good indicator of the overall capability of the modeling approach. In addition, the demonstration of the ability to apply the methods to the detection task of the binary classification scenario showed interesting results on the train set, but could not reach an adequate performance when the fitted model was applied on the test set.

Like displayed in CRISP-DM, the results of the evaluation influence one of the earliest stages, „Business Understanding“, to outline achievable goals of the project. In the case of this thesis, the results and perspectives are shown and set in context of the performance capabilities for the defined use cases. Especially for future work, an increase in the available size of the dataset would probably be one of the easiest steps to introduce, providing the highest potential on the outcome regarding the results.

The process chain of predicting/classifying a substance is based on the assumption of pre-selecting the explosives out of all available analytes in the dataset. This can be done by applying the already existing classifier for this dataset, developed and published independently from this work, introduced in section 3.3. Therefore, the developed ML pipeline of this thesis can be implemented in a present decision-making framework.

Regarding the training/fitting procedure of the models chosen, there may be room for improvement, especially in terms of the Grid Search procedure. Several more time efficient methods exist, that could be used instead, since the optimization procedure for the hyperparameter search is independent of the ML models selected. However, no increase in performance could be achieved as the implemented Grid Search already covers the complete search space.

For the evaluation of the hyperparameters, most of the time, the V-Measure has shown its potential as a reliable indicator for the clustering performance. The trade-off between homogeneity and completeness serves as a good insight about the quality of the clustering partition. All remaining metrics were mostly used as a fallback.

6 | Outlook and conclusion

The aim of this thesis was to develop a framework for processing sensor data to be able to classify unknown substances based on their chemical structure and energetic potential. For this purpose, the necessary prerequisite steps have been shown, regarding the data preparation and converting the raw data into a more usable (data) format. In addition, the need of preprocessing has been demonstrated in context of the various sensor responses, e. g. the interpolation or downsampling steps. Especially the averaging method introduced for the implementation of the new proposed framework differs greatly from the procedures applied in already existing publications regarding this dataset. Several approaches of working with this data are possible. Especially for the preparation and pre-processing stage, the chosen methods can vary greatly in their implementation and are not finalized. Depending on the context of use and the time requirements, even better results may be obtained for the data cleaning stage of the sensor signals. For the application of the methods chosen in this thesis, a suitable set of tools could be found, which enabled a satisfying quality of the results for the processed data.

A large contribution in this thesis was based on the development of an alternative way for extracting features from sensor data without relying too heavily on domain knowledge. In this context, the BOSS transformation has been applied on the TS data. One of the greater advantages of this method is the ability to be able to regulate the degree of noise reduction as well as complexity of the extracted pattern and obtained feature set. This allows for an exploratory approach to be able to find the most suitable hyperparameters for the given sensor data. The use of this approach has enabled several new opportunities. One of the biggest advantages is the representation of TS data through the BoW model. Many ML approaches have been developed to work with this kind of discrete data structure, one of them being LDA, which was applied in the context of this thesis. The LDA algorithm has a comparatively high degree of flexibility, while transforming the feature set of the BOSS model into a lower dimensional space. In combination with the KM algorithm, a search procedure has been performed to find the best set of hyperparameters in combination for both models. In conjunction with external class labels, given the specific use cases, the unsupervised approach could be leveraged as a classifier, using the resulting clusters as predictions for the class membership.

The results obtained through this ML pipeline were mixed. For the use case of classifying analytes according to their energetic potential, a good performance could be achieved. Regarding the identification of analytes based on their chemical structure, a weaker performance was shown. Likewise for the additional test of the capability

of detecting explosive analytes, a good model performance could only be achieved on the training set, but applied on the test set, the predictions were not reliable. As an outlook, there might still be a lot of room for improvement regarding the capabilities of the proposed framework. As for the use case of chemical structures, increasing the sample size of the available dataset might already lead to a significant improvement. Regarding the detection of explosives, in addition, it could be beneficial to balance the train set with an almost equal ratio of benign to explosive analytes, and avoid any disadvantages of an unbalanced dataset in terms of the application of ML concepts.

Finally, in regards to the „Deployment“ stage of CRISP-DM, various ways exist for implementing the proposed framework into the decision process for a real-life scenario. As mentioned in previous sections, the application for the two use cases is intended after the detection of explosives took place. An already existing solution, developed for the available dataset in this thesis, is able to achieve reliable results in terms of the binary classification scenario. Therefore, the creation of new ML models was focused on the provided use cases and the explosive substances only. The deployment of the framework can be realized in an offline setting. The training/fitting of the models will be done with the available data. Any predictions for new data will be done in an identical offline setting, requiring the pre-processing pipeline as well as the fitted feature transformation for every new observation.

Bibliography

Abanda, A.; U. Mori & J. A. Lozano (2019). „A Review on Distance-Based Time Series Classification“. In: *Data Mining and Knowledge Discovery* 33.2, pp. 378–412. DOI: 10.1007/s10618-018-0596-4.

Aggarwal, C. C. & C. K. Reddy (2014). *Data Clustering: Algorithms and Applications*. Data Mining and Knowledge Discovery Series. CRC Press. ISBN: 978-1-4665-5822-9.

Arthur, D. & S. Vassilvitskii (2007). „K-Means++: The Advantages of Careful Seeding“. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '07. New Orleans, USA: Society for Industrial and Applied Mathematics, pp. 1027–1035. URL: https://dl.acm.org/doi/10.5555/1283383.1283494.

Bagnall, A.; J. Lines; A. Bostrom; J. Large & E. Keogh (2017). „The Great Time Series Classification Bake Off: A Review and Experimental Evaluation of Recent Algorithmic Advances“. In: *Data Mining and Knowledge Discovery* 31.3, pp. 606–660. DOI: 10.1007/s10618-016-0483-9.

Bennett, B. T. (2018). *Understanding, Assessing, and Responding to Terrorism. Protecting Critical Infrastructure and Personnel*. 2nd ed. Wiley. ISBN: 978-1-119-23781-5.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer-Verlag. ISBN: 0-521-86571-9.

Blei, D. M.; A. Y. Ng & M. I. Jordan (2003). „Latent Dirichlet Allocation“. In: *Journal of Machine Learning Research* 3, pp. 993–1022. URL: https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf (last accessed June 1, 2021).

Bracewell, R. N. (2000). *The Fourier Transform and Its Applications*. 3rd ed. Electrical Engineering Series. McGraw-Hill Book Co. ISBN: 0-07-116043-4.

Bui, Q. V.; K. Sayadi; S. B. Amor & M. Bui (2017). „Combining Latent Dirichlet Allocation and K-Means for Documents Clustering: Effect of Probabilistic Based Distance Measures“. In: *Proceedings of the 9th Asian Conference on Intelligent Information and Database Systems*. ACIIDS '17. Kanazawa, Japan: Springer-Verlag, pp. 248–257. DOI: 10.1007/978-3-319-54472-4_24.

Chapman, P.; J. Clinton; R. Kerber; T. Khabaza; T. Reinartz; C. Shearer & R. Wirth (2000). *CRISP-DM 1.0: Step-by-Step Data Mining Guide*. The CRISP-DM Consortium.

Chen, Y. & B. Qi (2019). „Representation Learning in Intraoperative Vital Signs for Heart Failure Risk Prediction“. In: *BMC Medical Informatics and Decision Making* 19.1, p. 260. DOI: 10.1186/s12911-019-0978-6.

Cuturi, N. & M. Blondel (2017). „Soft-DTW: A Differentiable Loss Function for Time-Series“. In: *Proceedings of the 34th International Conference on Machine Learning*. ICML '17. Sydney, Australia: PMLR, pp. 894–903. URL: https://dl.acm.org/doi/10.5555/3305381.3305474.

Faouzi, J. & H. Janati (2020). „pyts: A Python Package for Time Series Classification“. In: *Journal of Machine Learning Research* 21.46, pp. 1–6. URL: https://www.jmlr.org/papers/volume21/19-763/19-763.pdf (last accessed June 19, 2021).

Fourier, J.-B. J. (1822). *Théorie analytique de la chaleur*. Paris, France: Firmin Didot.

Fowlkes, E. B. & C. L. Mallows (1983). „A Method for Comparing Two Hierarchical Clusterings“. In: *Journal of the American Statistical Association* 78.383, pp. 553–569. DOI: 10.1080/01621459.1983.10478008.

Fulcher, B. D. & N. S. Jones (2014). „Highly Comparative Feature-Based Time Series Classification“. In: *IEEE Transactions on Knowledge and Data Engineering* 26.12, pp. 3026–3037. DOI: 10.1109/TKDE.2014.2316504.

Giorgino, T. (2009). „Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package“. In: *Journal of Statistical Software* 31.7, pp. 1–24. DOI: 10.18637/jss.v031.i07.

Grubbs, F. E. (1950). „Sample Criteria for Testing Outlying Observations“. In: *The Annals of Mathematical Statistics* 21.1, pp. 27–58. DOI: 10.1214/aoms/1177729885.

Guaman, A. V.; P. Lopez & J. Torres-Tello (2019). „Multivariate Discrimination Model for TNT and Gunpowder Using an Electronic Nose Prototype: A Proof of Concept“. In: *Proceedings of the 2019 International Conference on Information Technology & Systems*. ICITS '19. Quito, Ecuador: Springer-Verlag, pp. 284–293. DOI: 10.1007/978-3-030-11890-7_28.

Guts, Y. (2018). *Target Leakage in Machine Learning*. 5th International Conference on Artificial Intelligence and Data Science Applications. AI Ukraine '18. Kyiv, Ukraine: AltexSoft. URL: https://aiukraine.com/wp-content/uploads/2018/09/12_00-Yuriy-Guts-Target-Leakage-in-Machine-Learning-.pdf (last accessed May 22, 2021).

Halkidi, M.; Y. Batistakis & M. Vazirgiannis (2001). „On Clustering Validation Techniques“. In: *Journal of Intelligent Information Systems* 17.2, pp. 107–145. DOI: 10.1023/A:1012801612483.

Hastie, T.; R. Tibshirani & J. Friedman (2009). *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*. 2nd ed. Springer Series in Statistics. Springer-Verlag. ISBN: 978-0-387-84858-7.

Hoffman, M. D.; D. M. Blei; C. Wang & J. Paisley (2013). „Stochastic Variational Inference“. In: *Journal of Machine Learning Research* 14.4, pp. 1303–1347. URL: https://jmlr.org/papers/volume14/hoffman13a/hoffman13a.pdf (last accessed June 1, 2021).

Hsieh, T.-Y.; S. Wang; Y. Sun & V. Honavar (2021). „Explainable Multivariate Time Series Classification: A Deep Neural Network Which Learns to Attend to Important Variables As Well As Time Intervals“. In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. WSDM '21. Virtual Event, Israel: Association for Computing Machinery, pp. 607–615. DOI: 10.1145/3437963.3441815.

Hubert, L. & P. Arabie (1985). „Comparing Partitions“. In: *Journal of Classification* 2.1, pp. 193–218. DOI: 10.1007/BF01908075.

Jain, A. K. (2010). „Data Clustering: 50 Years Beyond K-Means“. In: *Pattern Recognition Letters* 31.8, pp. 651–666. DOI: 10.1016/j.patrec.2009.09.011.

Jain, A. K. & R. C. Dubes (1988). *Algorithms for Clustering Data*. Prentice Hall Advanced Reference Series. Prentice-Hall. ISBN: 0-13-022278-X.

James, G.; D. Witten; T. Hastie & R. Tibshirani (2013). *An Introduction to Statistical Learning. With Applications in R*. Springer Texts in Statistics. Springer-Verlag. ISBN: 978-1-4614-7138-7.

Jelodar, H.; Y. Wang; C. Yuan; X. Feng; X. Jiang; Y. Li & L. Zhao (2019). „Latent Dirichlet Allocation (LDA) and Topic Modeling: Models, Applications, a Survey“. In: *Multimedia Tools and Applications* 78.11, pp. 15169–15211. DOI: 10.1007/s11042-018-6894-4.

Konstantynovski, K. (2018). „Kleine Helfer für große Herausforderungen: Entwicklung, Bewertung und Anwendung von Halbleitergassensoren zur Detektion von energetischen Materialien“. PhD thesis. Universität zu Köln, Mathematisch-Naturwissenschaftliche Fakultät. URL: https://nbn-resolving.de/urn:nbn:de:hbz:38-79533.

Konstantynovski, K.; C. Hammer; G. Njio; N. Wenzel; G. Holl & T. M. Klapotke (n. d.). „Library Free Bulk Detection of Explosives – Combining Simple Sensors for Resolving a Complicated Issue“. Unpublished (received on Oct. 2020).

Konstantynovski, K.; G. Njio; F. Börner; A. Lepcha; T. Fischer; G. Holl & S. Mathur (2018). „Bulk Detection of Explosives and Development of Customized Metal Oxide Semiconductor Gas Sensors for the Identification of Energetic Materials“. In: *Sensors and Actuators B: Chemical* 258, pp. 1252–1266. DOI: 10.1016/j.snb.2017.11.116.

Konstantynovski, K.; G. Njio & G. Holl (2017). „Detection of Explosives – Studies on Thermal Decomposition Patterns of Energetic Materials by Means of Chemical

and Physical Sensors“. In: *Sensors and Actuators B: Chemical* 246, pp. 278–285. DOI: 10.1016/j.snb.2017.02.077.

Kuncheva, L. I. & S. T. Hadjitodorov (2004). „Using Diversity in Cluster Ensembles“. In: *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics*. ICSMC ’04. The Hague, Netherlands: IEEE, pp. 1214–1219. DOI: 10.1109/ICSMC.2004.1399790.

Lin, J.; E. Keogh; L. Wei & S. Lonardi (2007). „Experiencing SAX: A Novel Symbolic Representation of Time Series“. In: *Data Mining and Knowledge Discovery* 15.2, pp. 107–144. DOI: 10.1007/s10618-007-0064-z.

Lloyd, S. (1982). „Least Squares Quantization in PCM“. In: *IEEE Transactions on Information Theory* 28.2, pp. 129–137. DOI: 10.1109/TIT.1982.1056489.

Manning, C. D.; P. Raghavan & H. Schütze (2009). *An Introduction to Information Retrieval*. Cambridge University Press. ISBN: 0-521-86571-9.

Martínez-Plumed, F.; L. Contreras-Ochando; C. Ferri; J. Hernández Orallo; M. Kull; N. Lachiche; M. J. Ramírez Quintana & P. A. Flach (2019). „CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories“. In: *IEEE Transactions on Knowledge and Data Engineering (Early Access)*. DOI: 10.1109/TKDE.2019.2962680.

Maurer, S.; R. Makarow; J. Warmer & P. Kaul (2015). „Fast Testing for Explosive Properties of mg-scale Samples by Thermal Activation and Classification by Physical and Chemical Properties“. In: *Sensors and Actuators B: Chemical* 215, pp. 70–76. DOI: 10.1016/j.snb.2015.03.045.

McCaffrey, J. (2015). „Test Run – K-Means++ Data Clustering“. In: *msdn magazine* 30 (8), pp. 62–68. URL: https://download.microsoft.com/download/e/2/a/e2aca573-591d-4dc9-afb9-e6260d3c6046/mdn_0815dg.pdf (last accessed June 7, 2021).

McLaurin, E.; A. D. McDonald; J. D. Lee; N. Aksan; J. Dawson; J. Tippin & M. Rizzo (2014). „Variations on a Theme: Topic Modeling of Naturalistic Driving Data“. In: *Proceedings of the 58th Human Factors and Ergonomics Society Annual Meeting*. Vol. 58. HFES ’14 1. San Diego, USA: Human Factors & Ergonomics Society, pp. 2107–2111. DOI: 10.1177/1541931214581443.

Müller, M. (2007). „Dynamic Time Warping“. In: *Information Retrieval for Music and Motion*. Springer-Verlag, pp. 69–84. ISBN: 978-3-540-74048-3.

Nisbet, R.; G. Miner & K. Yale (2018). *Handbook of Statistical Analysis and Data Mining Applications*. 2nd ed. Elsevier. ISBN: 978-0-12-416645-5.

Pedregosa, F.; 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 & E. Duchesnay (2011). „Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830. URL: https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf (last accessed June 22, 2021).

Petitjean, F.; A. Ketterlin & P. Gançarski (2011). „A Global Averaging Method for Dynamic Time Warping, with Applications to Clustering". In: *Pattern Recognition* 44.3, pp. 678–693. DOI: 10.1016/j.patcog.2010.09.013.

Rand, W. M. (1971). „Objective Criteria for the Evaluation of Clustering Methods". In: *Journal of the American Statistical Association* 66.336, pp. 846–850. DOI: 10.1080/01621459.1971.10482356.

Reimann, P. & A. Schütze (2013). „Sensor Arrays, Virtual Multisensors, Data Fusion, and Gas Sensor Data Evaluation". In: *Gas Sensing Fundamentals*. Ed. by C. D. Kohl & T. Wagner. Vol. 15. Springer Series on Chemical Sensors and Biosensors (Methods and Applications). Springer-Verlag, pp. 67–107. DOI: 10.1007/5346_2013_52.

Rohit, K. J. (2016). „Using Dynamic Time Warping Distances as Features for Improved Time Series Classification". In: *Data Mining and Knowledge Discovery* 30.2, pp. 283–312. DOI: 10.1007/s10618-015-0418-x.

Romano, S.; N. X. Vinh; J. Bailey & K. Verspoor (2016). „Adjusting for Chance Clustering Comparison Measures". In: *Journal of Machine Learning Research* 17.1, pp. 4635–4666. URL: https://jmlr.org/papers/volume17/15-627/15-627.pdf (last accessed June 10, 2021).

Rosenberg, A. & J. Hirschberg (2007). „V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. EMNLP-CoNLL '07. Prague, Czech Republic: Association for Computational Linguistics, pp. 410–420. URL: https://www.aclweb.org/anthology/D07-1043.pdf (last accessed June 10, 2021).

Ruiz, A. P.; M. Flynn; J. Large; M. Middlehurst & A. Bagnall (2021). „The Great Multivariate Time Series Classification Bake Off: A Review and Experimental Evaluation of Recent Algorithmic Advances". In: *Data Mining and Knowledge Discovery* 35.2, pp. 401–449. DOI: 10.1007/s10618-020-00727-3.

Schäfer, P. (2015a). „Scalable Time Series Similarity Search for Data Analytics". PhD thesis. Humboldt University of Berlin, Faculty of Mathematics and Natural Sciences. DOI: 10.18452/17338.

Schäfer, P. (2015b). „The BOSS Is Concerned with Time Series Classification in the Presence of Noise“. In: *Data Mining and Knowledge Discovery* 29.6, pp. 1505–1530. DOI: 10.1007/s10618-014-0377-7.

Schäfer, P. & M. Högqvist (2012). „SFA: A Symbolic Fourier Approximation and Index for Similarity Search in High Dimensional Datasets“. In: *Proceedings of the 15th International Conference on Extending Database Technology*. EDBT '12. Berlin, Germany: Association for Computing Machinery, pp. 516–527. DOI: 10.1145/2247596.2247656.

Shumway, R. H. & D. S. Stoffer (2017). *Time Series Analysis and Its Applications. With R Examples*. 4th ed. Springer Texts in Statistics. Springer-Verlag. ISBN: 978-3-319-52451-1.

Sievert, C. & K. Shirley (2014). „LDAvis: A method for Visualizing and Interpreting Topics“. In: *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*. ACL '14. Baltimore, USA: Association for Computational Linguistics, pp. 63–70. URL: https://nlp.stanford.edu/events/illvi2014/papers/sievert-illvi2014.pdf (last accessed July 13, 2021).

Tan, P.-N.; M. Steinbach; V. Kumar & A. Karpatne (2019). *Introduction to Data Mining*. 2nd ed. Pearson. ISBN: 0-133-12890-3.

Tavenard, R.; J. Faouzi; G. Vandewiele; F. Divo; G. Androz; C. Holtz; M. Payne; R. Yurchak; M. Rußwurm; K. Kolar & E. Woods (2020). „Tslearn, a Machine Learning Toolkit for Time Series Data“. In: *Journal of Machine Learning Research* 21.118, pp. 1–6. URL: https://jmlr.org/papers/volume21/20-091/20-091.pdf (last accessed June 19, 2021).

Twinandilla, S.; S. Adhy; B. Surarso & R. Kusumaningrum (2018). „Multi-Document Summarization Using K-Means and Latent Dirichlet Allocation (LDA) – Significance Sentences“. In: *Procedia Computer Science* 135, pp. 663–670. DOI: 10.1016/j.procs.2018.08.220.

Vinh, N. X.; J. Epps & J. Bailey (2010). „Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance“. In: *Journal of Machine Learning Research* 11.95, pp. 2837–2854. URL: https://jmlr.org/papers/volume11/vinh10a/vinh10a.pdf (last accessed June 10, 2021).

Wickham, H. (2014). „Tidy Data“. In: *Journal of Statistical Software* 59.10, pp. 1–23. DOI: 10.18637/jss.v059.i10.

Wu, J.; H. Xiong & J. Chen (2009). „Adapting the Right Measures for K-Means Clustering“. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. Paris, France: Association for Computing Machinery, pp. 877–886. DOI: 10.1145/1557019.1557115.

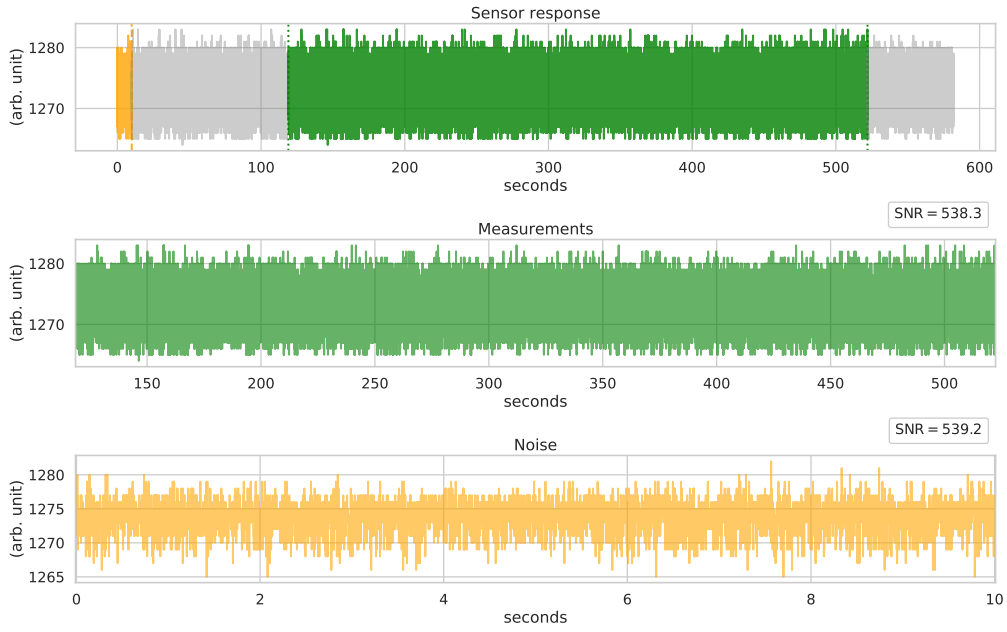
Appendix

A – Tables

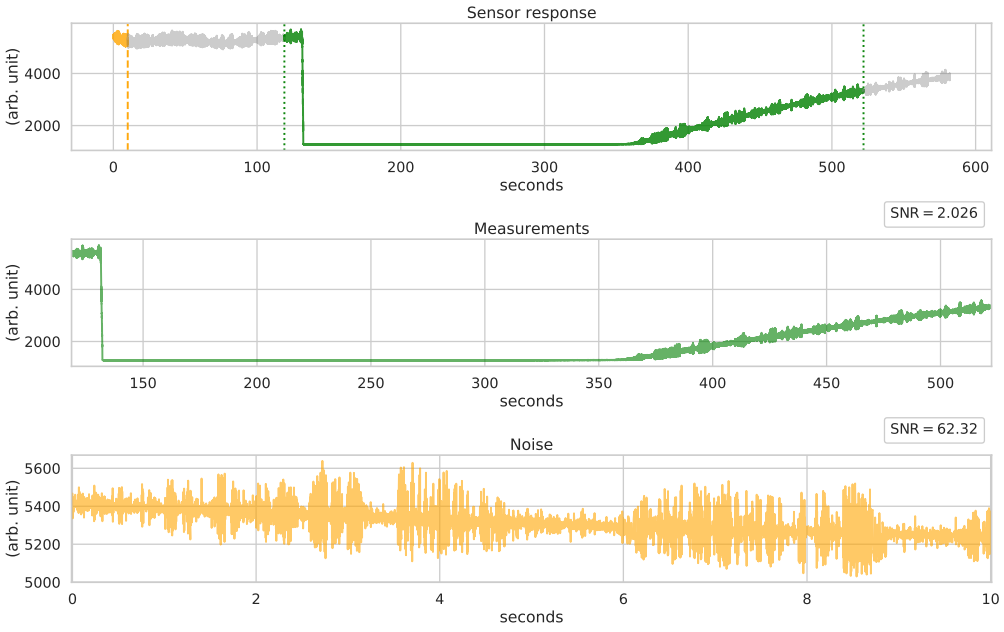
Tab. A.1: Listing of all analytes.‡ The table represents the train *(left)* and test set *(right)*. The top part contains the explosives; the bottom are benign substances.
‡ Table taken from Konstantynovski *et al.* (n. d.).

Train			Test		
Nr.	Name	Abbr.	Nr.	Name	Abbr.
1.	ammonium nitrate	AN	1.	5,5-bis(2,4,6-trinitrophenyl)-2,2-bis(1,3,4-oxadiazole)	TKX-55
2.	hexamethylene triperoxide diamine	HMTD	2.	bis(3,4,5-trinitropyrazol-1-yl) methane	BTNPM
3.	cyclotrimethylene trinitramine	RDX	3.	2,6-diamino-3,5-dinitro pyrazin-1-oxide	LLM 105
4.	pentaerythritol tetranitrate	PETN	4.	hexanitro isowurtzitane	CL-20
5.	Semtex 1A	Sem	5.	dihydroxylammonium 5,5-bistetrazol-1,1-diolate	TKX-50
6.	triacetone triperoxide	TATP	6.	potassium 1,5-di(nitramino)-tetrazole	K_2-DNAT
7.	trinitro phenyl methyl nitramine	Tet	7.	dihydroxylammonium 5,5-bis(3-nitro-1,2,4-triazolat-1N-oxide)	MAD-X1
8.	black powder	BP	8.	1-amino-1-(1H-tetrazol-5-yl)-azoguanidine	Tetrazene
9.	trinitro toluene	TNT			
10.	2,4,6-trinitro aniline	Picramide			
11.	nitro glykole + ammonium nitrate	Geosit 3			
12.	methyl ammonium nitrate + ammonium nitrate	Tovex SE			
13.	2-ethyl-1,3,5-trinitro benzene	TNEB			
14.	urea	Urea	9.	sodium iodide	NaI
15.	abrasive cleaner	Abr	10.	glyoxime	Glyoxime
16.	hexamethylene tetramine	Urt			
17.	sucrose	Su			

B – Figures



(A)



(B)

Fig. B.1: Two samples producing different response patterns for UST-5333. The intervals of the first 10 seconds (step 1) and 119–522 seconds (step 6–9) are colored in orange and green, respectively. To measure the validity of a given sensor response, the SNR gets calculated and compared for each step interval. The bottom plot *(B)* shows a valid response for the first sample of the analyte Semtex; the top *(A)* is showing the first run of the analyte Geosit, indicating no valid response.